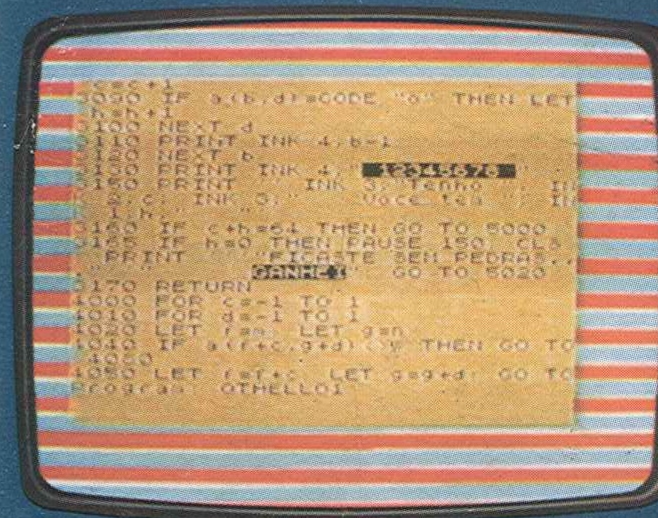
Este livro inclui uma gama de 26 programas completamente novos que vão desde os testes de memória aos jogos mais sofisticados como os de guerra. Os programas, aplicáveis aos mais elementares computadores domésticos, foram escritos numa versão BASIC muito fácil que o leitor poderá introduzir sem dificuldade em qualquer microcomputador. Cada programa é exaustivamente descrito e são sempre indicadas a lista de variáveis e a capacidade de memória requerida, incluindo também exemplos de execução.
Para além de muitas horas de entretenimento, este livro proporcionará ao leitor ideias novas que poderão ajudá-lo a melhorar os seus próprios programas.

EDITORIAL PRESENÇA

26 Programas Basic para Microcomputadores

157

DERRICK DAINES

26 PROGRAMAS BASIC

PARA MICROCOMPUTADORES

CULTURA E TEMPOS LIVRES

1. ABC do Xadrez, *Petar Trifunovitch* e *Sava Vukovitch*
4. ABC do Bridge, *Pierre Jais* e *H. Lahana*
5. Guia Prático de Fotografia, *W. D. Emanuel*
6. ABC do Judo, *E. J. Harrison*
7. Como Fazer Cinema, *Paul Petzold*
8. Bridge Moderno, *Pierre Jais* e *H. Lahana*
9. Fotografia — Técnicas e Truques I, *Edwin Smith*
10. ABC dos Estilos, da Arquitectura ao Mobiliário, *A. Aussel*
11. Fotografia — Técnicas e Truques II, *Edwin Smith*
12. A Pesca Submarina, *António Ribera*
13. Teoria dos Finais de Partida, *Yuri Averbach*
14. Aprenda Rádio, *B. Fighiera*
15. Guia do Cão, *Louise Laliberté-Robert e Jean-Pierre Robert*
16. ABC do Aquário, *Anthony Evans*
17. Iniciação à Electricidade e Electrónica, *Fernand Huré*
18. Os Transístores, *Fernand Huré*
19. Karaté I, *Albrecht Pflüger*
20. Iniciação ao Radiocomando dos Modelos Reduzidos, *C. Péricone*
21. Construa o seu Receptor, *B. Fighiera*
22. Montagens Electrónicas, *B. Fighiera*
23. O Berbequim Eléctrico, *Villy Dreier*
24. Cactos, *J. Nilaus Jensen*
25. Iniciação à Alta Fidelidade, *Peter Turner*
26. O Aquário de Água Doce, *Paulo de Oliveira*
27. ABC do Ténis, *Fonseca Vaz*
28. Karaté II, *Albrecht Pflüger*
29. ABC da Criação de Canários, *Curt Af Enehjelm*
30. Ginástica Feminina, *Sonja Helmer Jensen*
31. Cartomancia, *Thea Koch*
32. Calculadoras Electrónicas de Bolso, *E. Dam Ravn*
33. O Pastor Alemão, *Gilles Legrand*
34. Xadrez — Teoria do Meio Jogo I, *Bondarevsky*
35. Manual do Super 8 — I, *Myron A. Matzkin*
36. ABC da Criação de Periquitos, *Cyril H. Rogers*
37. O Livro dos Gatos, *Bärbel Gerber* e *Horst Bielfeld*
38. Manual do Super 8 — II, *Myron A. Matzkin*
39. ABC do Mergulho Desportivo, *Walter Mattes*
40. Circuitos Integrados/Aplicações Práticas, *F. Bergtold*
41. A Apicultura, *H. R. C. Riches*
42. ABC do Cultivo das Plantas, *H. G. Witham Fogg*
43. ABC da Criação de Pombos, *Kai R. Dahl*
44. Construção de Caixas Acústicas de Alta Fidelidade, *R. Brault*
45. Raças de Canários, *Klaus Speicher*
46. Jogos de Cartas, *Graciano Dolma*
47. Spaniels, *H. S. Lloyd*
48. ABC da Caça, *Fabián Abril*
49. Aprenda Televisão, *Gordon J. King*
50. Iniciação à Pesca, *Juan Nadal*
51. Basquetebol, *Marius Norregard*
52. Cães de Caça, *Santiago Pons*
53. Aprenda Electrónica, *T. L. Squires* e *C. M. Deason*
54. A Avicultura, *Jim Worthington*
55. A Produção de Coelhos, *P. Surdeau* e *R. Henaff*
56. ABC dos Computadores, *T. F. Fry*
57. Natação para crianças, *John Idorn*
58. O Boxer, *Anni Mortensen*
59. Voleibol, *Ole Hansen* e *Per-Göran Person*
60. Iniciação à Vela, *Donald Law*
61. ABC da Filatelia, *Jacqueline Caurat*
62. A Pesca à Beira-Mar, *J.-M. Böelle* e *B. Doyen*
63. Enxerto de Árvores de Fruto, *Alejo Rigau*
64. A Cultura do Morangueiro, *Luís Alsina Grau*
65. Emissores-Receptores (Walkies-Talkies), *P. Duranton*
66. Iniciação à Fotoelectrónica, *Heinz Richter*
67. Doces e Conservas de Fruta, *Robin Howe*
68. A Criação de Hamsters, *C. F. Snow*
69. A Criação de Porcos, *Roy Genders*
70. Calendário do Horticultor, *Luís Alsina Grau*
71. Jogos Electrónicos, *F. G. Rayer*
72. Cultivo de Cogumelos e Trufas, *Alejo Rigau*
73. Aprenda Televisão a Cores, *Gordon J. King*
74. Gravação em Fita Magnética, *Ian R. Sinclair*
75. Poda de Árvores e Arbustos, *Roy Genders*
76. Como Treinar o Seu Cão, *E. Fitch Daglish*
77. Instrumentos de Medida e Verificação, *Heinrich Stöckle*
78. A Criação de Caracóis, *Matías Josa*
79. Rádio — Fundamentos e Técnicas, *Gordon J. King*
80. Como Fazer Gelados, *Sylvie Thiébault*
81. Iniciação à Jardinagem, *Noel Clarasó*
82. A Congelação dos Alimentos, *Suzanne Lapointe*
83. Windfsurf — Prancha à Vela, *Ernstfried Prade*
84. Raças de Cães, *O. Hasselfeldt*
85. Rummy e Canasta, *Claus D. Grupp*
86. A Encadernação, *Annie Persuy*
87. Aprenda Electricidade, *Heinz Richter*
88. Taxidermia, Embalsamamento de Aves e Mamíferos, *Harry Hjortaa*
89. Jogging — Correr para Manter a Forma, *Werner Sonntag*
90. ABC da Cozinha Chinesa, *Sonya Richmond*
91. Jogos T.V., *C. Tavernier*
92. Amplificadores de Som, *Richard Zierl*
93. O Livro do Poker, *Claus D. Grupp*
94. Aprenda a Desenhar, *Rose-Marie de Prémont* e *Nicole Philippi*
95. O Minitrampolim na Escola, *Sonja Helmer Jensen* e *Klaus Dano*
96. Jogos de Luzes e Efeitos Sonoros para Guitarras, *B. Fighiera*
97. O Cultivo do Tomate, *Louis N. Flawn*
98. Pilhas Solares, *F. Juster*
99. A Criação Doméstica de Coelhos, *C. F. Snow*
100. Iniciação ao Futebol, *Wieland Männle* e *Heinz Arnold*
101. Horóscopos Chineses, *Georg Haddenbach*
102. Guia Prático de Marcenaria, *Charles H. Hayward*
103. Andebol, *Fritz* e *Peter Hattig*
104. Dispositivos Anti-Roubo, *H. Schreiber*
105. Perus, Pintadas e Codornizes, *Jérome Sauze*
106. Crepes — Doces e Salgados, *Florence Arzel*

107. Aperitivos e Entradas, *Myrette Tiano*
108. Ténis de Mesa, *Leslie Woollard*
109. Aprenda Surf, *R. Abbott* e *M. Baker*
110. Futebol — Técnica e Táctica, *Kurt Lavall*
111. A Vaca Leiteira, *Colin T. Whittemore*
112. O Cubo Mágico, *Josef Trajber*
113. O Perdigueiro Português, *José M. Correia*
114. Pizzas e Massas à Italiana, *Marieanne Ränk*
115. O Cubo Para Quem Já o Faz, *Josef Trajber*
116. A Pirâmide Mágica, A Torre, O Barril do Diabo, *M. Mrowka-W. J. Weber*
117. Gansos e Patos, *Marie Mourthe*
118. Iniciação ao Kung-Fu, *A. P. Harrington*
119. Electrónica e Fotografia, *Hanns-Peter Siebert*
120. O Livro da Fortuna, *Douglas Hill*
121. Construção de um Alimentador de corrente, *Waldemar Baitinger*
122. Hóquei em Patins, *Francisco Velasco*
123. Técnicas de Tiro, *Anton Kovacic*
124. Aprenda a Tricotar, *Uta Mix*
125. ABC da Patinagem, *Christa-Maria* e *Richard Kerler*
126. A Pesca e os seus Segredos, *Armand Deschamps*
127. O Osciloscópio, *R. Rateau*
128. Guia Prático da Banda do Cidadão, *T. M. Normand*
129. Sumos e Batidos, *Manfred Donderski*
130. Introdução à Programação de Microcomputadores, *Peter C. Sanderson*
131. Aprenda Croché, *Uta Mix*
132. ABC do Microprocessador, *P. Mélusson*
133. Guia Prático de Basic, *Roger Hunt*
134. Introdução à Electrónica Digital, *Ian Sinclair*
135. ABC do Vídeo, *David K. Matthewson*
136. Fotografia em Movimento, *Don Morley*
137. Guia de Cobol, *Ray Welland*
138. Fotografia a Pequena Distância, *Sidney F. Ray*
139. Guia Moderno da Canaricultura, *Manuel Gonçalves*
140. Minielectrónica para Amadores, *Heinz Richter*
141. ABC da Programação de Computadores, *John Shelley*
142. Tarot — O Futuro Pelas Cartas, *Edwin J. Nigg*
143. ABC da Equitação, *Dorothy Johnson*
144. Como Programar o Seu ZX 81, *Patrick Gueulle*
145. 100 Avarias TV e a Maneira Prática de as Detectar, *P. Duranton*
146. ABC da Horticultura, *Louis Giordano*
147. Basic Para Microcomputadores, *A. P. Stephenson*
148. Como Programar o seu ZX Spectrum, *Tim Hartnell* e *Dilwyn Jones*
149. Iniciação aos Motores Diesel, *David S. Maclean*
150. 60 Jogos para o ZX Spectrum, *David Harwood*
151. As Linhas da Mão, *Rosé Hubert*
152. Cozinha Italiana, *Rotraud Degner*
153. Manual do ZX Spectrum, *Simpson* e *Terrel*
154. 80 Assembler para o ZX Spectrum, *João Paulo Fragoso*
155. Aeróbica, *H. Schulz*
156. ABC do Atletismo, *Denis Watts*
157. 26 Programas Basic para Microcomputadores, *Derrick Daines*

DERRICK DAINES

26 PROGRAMAS BASIC PARA MICROCOMPUTADORES

Título original:
26 BASIC PROGRAMS FOR YOUR MICRO

Tradução de Conceição Jardim e Eduardo Nogueira
Capa de Rogério Silva

PREFÁCIO

O leitor tem perante si um livro único sobre programas de computadores — contendo programas que não pode encontrar em nenhum outro local. Apenas dois, «Barco a motor» e «Algarismos significativos» foram já impressos. Nestas condições, o livro não é apenas mais uma colecção de programas muito conhecidos mas sim um texto completamente novo, com jogos e ideias que nunca viram anteriormente a luz do dia. Os programas são todos meus, pelo que se quiser culpar alguém pode apontar-me o dedo...

Os programas podem ser classificados de diferentes maneiras. O método que adoptei é um tanto empírico, baseando-se numa apreciação subjectiva do seu comprimento e da sua complexidade. Não há dúvida de que este método produz algumas anomalias, mas espero que estas não retirem prazer ao utilizador.

Os programas variam desde o bastante simples ao extremamente complexo. Os primeiros podem ser traduzidos para qualquer computador, mesmo o mais simples, que utilize a BASIC como linguagem de programação, enquanto que os últimos necessitarão já de uma memória utilizável até 13 K. Incluí nas notas de cada programa alguns comentários sobre os requisitos especiais da conversão para a sua máquina.

Foram incluídas igualmente algumas notas de interesse para fazer enxertos noutros programas. Chamo a atenção do leitor para quaisquer pontos de cada programa que justifiquem um maior estudo do leitor com vista à inclusão noutros programas. Deste modo, espero que o prazer que tenho ao programar um computador possa ser passado a outros, e dar o meu pequeno contributo para que o hobby a que nos dedicamos progrida.

D. D.

INTRODUÇÃO

Tende a ser aceite que a «melhor» ou «mais natural» versão da BASIC é aquela que por acaso é utilizada pela nossa máquina. Como possuidor e utilizador de bastantes versões, preferi escrever em STWP BASIC porque é muito fácil de traduzir para qualquer outra BASIC. Todos os programas deste livro foram escritos nesta versão, utilizando um computador SWTP 6800 com uma memória de 32 K, e gravados em «minifloppies» (disquettes). Poucas instruções dependem com efeito da máquina usada (mas apresento uma lista das que o são: algumas podem muito simplesmente ser omitidas). O que se segue é uma descrição da SWTP BASIC para diminuir as dificuldades que o leitor possa encontrar na tradução.

VARIÁVEIS — Uma variável pode ser qualquer letra isolada, ou uma letra e um número entre 0 e 9.

VARIÁVEIS DE "STRING" — Podem ser designadas por qualquer letra seguida do sinal $, e são usadas para guardar na memória dados literais. Por exemplo, A$ pode ser "1224" ou "Exemplo", 1224 (as aspas definem a string). Quando as strings são inicializadas, são dimensionadas para um máximo de 32 caracteres desta BASIC: é no entanto possível ampliar a string usando a ordem STRING, por exemplo STRING = 50. As linhas de instruções são limitadas a 72 caracteres. São aceites linhas de instruções múltiplas desde que estas sejam separadas usando o sinal ':'.

CONCATENAÇÃO — As strings podem ser concatenadas (juntadas entre si) recorrendo ao sinal '+'. Por exemplo, A$ = "Olá": B$ = "João": C$ = A$ + B$ (Neste caso C$ conterá a string "OláJoão", por não ter sido previsto um espaço de separação nem em A$ nem em B$).

ARRAYS — Se não tiverem sido dimensionados por uma instrução DIM, a máquina considera que o array ou quadro possui 10 elementos. O mesmo acontece com matrizes — se não forem dimensionadas, a máquina considera que se trata de uma matriz de 10×10. Para poupar memória, portanto, é aconselhável dimensionar os arrays. As variáveis de string devem igualmente ser dimensionadas como array.

VALORES INICIAIS — Quando se utilizam quaisquer variáveis pela primeira vez, a SWTP BASIC considera que possuem o valor inicial de 0. Não é necessário portanto atribuir-lhes esse valor, ao contrário do que acontece noutras versões desta linguagem.

LINE = X — Esta ordem pode ser usada para definir o número de posições de impressão numa linha, sendo X o número de posições desejado. Se a posição de impressão se encontra nos últimos 25% do comprimento da linha, e a máquina encontra um espaço, a SWTP BASIC força um retorno de linha e a passagem à linha seguinte, evitando assim que as palavras sejam cortadas no final da linha.

DATA — Estas instruções podem ser colocadas em qualquer ponto do programa. Usam-se ordens para separar os dados, mas não necessariamente no final de uma linha. Os dados numéricos e em string podem ser misturados, mas o programa deve chamá-los pela ordem correcta. As strings *não* necessitam de estar encerradas dentro de aspas a menos que incluam uma vírgula, caso em que as aspas são obrigatórias.

RESTORE — Leva o "pointer" de dados a indicar novamente a primeira instrução DATA.

END — Não necessária em SWTP BASIC a menos que se torne necessária para terminar um programa a meio da listagem. END pode aparecer mais do que uma vez, mas não é de modo algum necessário utilizá-la.

FOR... TO... NEXT... STEP — As instruções FOR e NEXT são usadas em conjunto para formar "loops". A variável presente na instrução FOR é redifinida de cada vez que executa o loop, num mínimo de uma vez. Quando encontra a instrução NEXT, a variável é incrementada de 1 se não tiver sido definido outro valor de STEP, e o programa retoma a execução da instrução que se segue a FOR... TO. Se ao ser incrementada a variável assumir um valor maior do que o indicado em TO, a execução passa para a linha que se segue à instrução NEXT. STEP pode

ser negativo, caso em que a execução continua até a soma ser *inferior* ao valor indicado em TO. FOR, NEXT e STEP podem ser expressões, mas são avaliadas apenas uma vez.

GOSUB — Depois de cumprida a subrotina, o controlo passa a execução para a linha que se segue à instrução GOSUB.

GOTO — Salto incondicional.

IF... THEN — Se a relação dada na instrução IF for "verdadeira", a execução passa para a linha que se segue a THEN. Por exemplo, IF X=5 THEN 50 transfere a execução para a linha 50 se e só se X=5. Se a relação é falsa, a execução passa para a linha que se segue à instrução IF — isto é, o resto dessa mesma linha não é considerado pela máquina, mesmo que se trate de uma linha com várias instruções.

É igualmente possível executar uma instrução BASIC válida depois de THEN. Por exemplo, IF X=5 PRINT "Correcto". Note igualmente que a SWTP BASIC parte do princípio de que THEN está presente na instrução. Algumas outras versões obrigam no entanto à inclusão desta palavra-chave.

INPUT — Permite ao utilizador comunicar à máquina dados ou strings. São permitidas diversas formas, como por exemplo INPUT X, Y, Z que permitirá três dados numéricos, ou INPUT Q$, R$ que pedirá duas strings. Os valores numéricos apresentados à máquina quando esta pede uma string são aceites e convertidos em strings.

As instruções PRINT e INPUT podem ser combinadas, como no seguinte exemplo: INPUT "Dê-me um número", N. Esta instrução imprime no visor a mensagem "Dê-me um número?" (note que acrescenta o sinal de interrogação) e espera que seja dada entrada a um valor numérico. Um outro aspecto a notar nesta instrução é a vírgula que se segue à string de mensagem; certas versões da BASIC utilizam aqui um ponto e vírgula.

LET — Esta instrução é opcional

ON... GOTO — ON... GOSUB — Estas instruções transferem a execução para os números de linha indicados a seguir a GOTO ou GOSUB, conforme o valor da expressão que se segue a ON. Por exemplo:

ON X + 1 GOTO 100, 200, 300, 400

Se X=0, a execução é transferida para a linha 100; se X=1, para a linha 200; se X=2 para a 300, e se X=3 para a 400. Qualquer outro valor de X produzirá uma mensagem de erro.

REM — Indica "remark" (observação), sendo apenas um auxiliar para os humanos. Estas linhas são omitidas pelas máquinas, pois a BASIC ignora todas as instruções REM.

Funções

ABS(X) — dá o valor absoluto de X; ou seja, nunca pode ter um valor negativo.

ATAN(X) — dá o ângulo em radianos cuja tangente é X.

ASC(STRING) — dá o valor decimal do primeiro caracter da string; ou seja, ASC("?") dá 63.

CHR$(X) — dá o caracter único equivalente ao valor decimal ASCII de X. Por exemplo, CHR$(65) dá A.

COS(X) — dá o co-seno de X, que deve ser indicado à máquina em radianos.

DEF FNX(Y) = (exp) — define uma função X designada por uma letra única. Permite a criação de qualquer função, por exemplo DEF FNA(Y) = 3.14*Y↑2. Y designa qualquer variável, podendo de facto ser substituída pelo valor que se quiser ao chamar a função. Por exemplo:

100 LET G = FNA(23)

dará a G o valor 1661,06 dado que Y foi subsituída por 23. Se na expressão existir uma variável diferente de Y, é tomado em conta o valor actual dessa variável ao avaliar a função.

EXP(X) — dá a base de logaritmos naturais elevados à potência X. O inverso desta função é LOG(X).

INT(X) — dá o maior número inteiro inferior a X, isto é, arredonda para menos.

LEFT$(X$,N) — dá uma string de caracteres com um comprimento N, começando pelo que se encontra mais à esquerda em X$.

RIGHT$(X$,N) — como acima, a partir do caracter mais à direita em X$.

MID$(X$,S,T) — extrai de X$ uma string de caracteres com o comprimento T, começando pelo que se encontra na posição S.

LEN(X$) — dá o número de caracteres que se encontram em X$, incluindo quaisquer espaços.

LOGx(X) — dá o logaritmo natural de X.

RND e RND(O) — dá um número aleatório entre 0 e 1.

SGN(X) — dá o sinal de X; isto é, −4 produz −1; +4 produz 1.

SIN(X) — dá o seno do ângulo X indicado em radianos.

SQR(X) — dá a raíz quadrada de X.

STR$(X) — traduz uma variável numérica numa string.

VAL(X$) — é o oposto: dá sob a forma de uma constante numérica o primeiro valor encontrado em X$.

TAB(X) — desloca a posição de impressão para a coluna X dentro da mesma linha. Pode ser uma expressão. Note que no caso de TAB obrigar a passar para a linha seguinte, ou a seguir a uma outra instrução PRINT, se torna necessário o emprego do separador ';' a seguir a TAB. Por exemplo, PRINT TAB (16);

TAN(X) — dá a tangente do ângulo X indicado em radianos.

Notas sobre os gráficos

Nas listagens que se seguem não foram incluídos quaisquer gráficos. Certamente muitos criticarão este facto, mas peço aos leitores que compreendam que todos os gráficos dependem bastante da máquina que se usa. A inclusão de gráficos restringiria portanto os possíveis leitores àqueles que possuem apenas uma determinada máquina, enquanto que a omissão dos gráficos permite o uso destas listagens pelos possuidores de qualquer máquina. Nestas condições, e sempre que possível, foram omitidas as instruções que variam de máquina para máquina e tentou-se manter as listagens bastante simples.

Não há dúvidas no entanto que quase todos os programas apresentados seriam melhorados incorporando gráficos ou algumas instruções especiais. Devo deixar no entanto este cuidado ao leitor interessado. É impossível num único livro dar sugestões ou soluções para todos os casos.

As poucas instruções variáveis em função da máquina que aqui se utilizam são restringidas à variedade CHR$(X), onde o X designa certas funções. Segue-se uma lista destas; o leitor deverá simplesmente substituí-las pelo X que executa a mesma função na sua máquina. Se não dispuser de uma função especial para o efeito, omita-a, o que em geral não terá efeitos no programa.

CHR$(5)	Desliga o cursor
CHR$(6)	Apaga o final da linha

CHR$(7)	Som
CHR$(8)	Cursor à esquerda
CHR$(9)	Cursor à direita
CHR$(10)	Cursor em baixo
CHR$(11)	Cursor em cima
CHR$(12)	Mudar página
CHR$(13)	Retorno ao início de linha
CHR$(14)	"Scrolling" (rolamento) da imagem
CHR$(15)	Não aplicável
CHR$(16)	"Home" (Envio do cursor para o canto superior esquerdo)
CHR$(17)	Leitor de fita ligado
CHR$(18)	"Punch" ligado
CHR$(19)	Leitor de fita desligado
CHR$(20)	"Punch" desligado
CHR$(21)	Cursor ligado
CHR$(22)	Apagar...
CHR$(23)	Não aplicável
CHR$(24)	Não aplicável
CHR$(25)	Inicializar o terminal
CHR$(26)	Modo página
CHR$(27)	Cursor normal
CHR$(28)	Cursor intermitente
CHR$(29)	Inverter visor

Rolamento do visor

Em modo página, quando o cursor atinge o final da última linha continua a imprimir no início da linha superior. Note que (pelo menos na minha máquina) não limpa automaticamente o visor; é necessário utilizar uma instrução nesse sentido, o que constitui uma característica bastante útil no caso de se pretender apenas alterar alguns caracteres aqui e ali. O mesmo acontece com a ordem "Home", que apenas reenvia o cursor para a posição no cimo à esquerda sem limpar o visor.

O paginamento é útil para gráficos automáticos, isto é, para mover caracteres. Isto é feito por exemplo no programa "Puzzle". Com o cursor desligado e o terminal no modo página, os caracteres parecem desligar no visor. Isto é feito imprimindo uma

string como, por exemplo, "ABCDE". Se for executado um retorno de linha neste momento e impresso a string "A BC D E", a letra C parecerá mover-se para a esquerda. É esta a base de muitos programas ditos "gráficos", incluindo os executados em computadores muito vendidos.

Em modo "scrolling", quando o cursor atinge o final da linha inferior todo o visor parece rolar para cima uma linha, apresentando uma linha nova na parte inferior, onde o cursor continua a imprimir. O efeito disto é semelhante ao de um rolo de papel saindo continuamente de uma impressora.

O interesse do scrolling depende bastante da velocidade do terminal. Se for muito lento (digamos 300 baud ou menos), a velocidade é quase tão aceitável como a de leitura normal. Se for muito rápido (por exemplo mais de 1200 baud) o visor enche-se num ou dois segundos, depois do que o programa pára a fim de permitir a leitura do que foi escrito. Entre estes extremos, o efeito de rolamento dos caracteres pode ser bastante irritante.

I

SOMA-DIFERENÇA (0,5 K)

Este programa, o mais simples do livro, poderá ser executado facilmente em qualquer computador que utiliza a linguagem BASIC, podendo ser traduzido facilmente para código-máquina no caso dos pequenos computadores que a não utilizem. Pode até ser traduzido para máquinas de calcular programáveis, desde que se omita o texto.

O jogo é evidentemente pensado apenas para crianças. É extremamente simples, mas se não se conhece o segredo pode ser bastante frustrante encontrar os dois números escondidos.

Para programadores principiantes, pode ser interessante observar que uma instrução PRINT pode incluir cálculo, como acontece nas linhas 110, 120 e 290. Não há necessidade de fazer o cálculo antes de imprimir a resposta, e o método apresentado pode poupar espaço de programa e de variáveis. Como é óbvio, se se quiser fazer mais alguma coisa ao resultado de um cálculo, é melhor fazer o cálculo anteriormente.

Note ainda que o programa não tem saída, continuando permanentemente a executar o loop. O programa só desaparece quando se desliga a máquina ou se utiliza uma ordem. Se quiser, pode reescrever a linha 300 a fim de perguntar ao utilizador se quer jogar de novo:

```
300 PRINT: PRINT "Carregue em 1 para jogar de novo";
310 INPUT A
320 IF A=1 THEN 10
330 END
```

Se puder utilizar strings e linhas com várias instruções, pode reduzir tudo isto a uma única linha:


```
300 PRINT: INPUT "Jogar de novo (s/n)?", A$:
  IF A$="s" THEN 10
```

Veremos muitos exemplos semelhantes a isto neste livro.

As linhas 30 e 40 são interessantes. A linha 120 exige que X seja sempre maior do que Y, mas no entanto as linhas 10 e 20 não garantem isto quando produzem os dois números secretos. Portanto, a linha 30 verifica qual é maior e, em caso de necessidade, a linha 40 troca os números, colocando o maior em X e o menor em Y.

Lista de variáveis

X	primeiro número secreto
Y	segundo número secreto
Z	armazenamento temporário
X1	adivinha do primeiro número pelo utilizador
Y1	adivinha o segundo número pelo utilizador
N	contagem do número de problemas colocados
R	contagem do número de problemas resolvidos

Listagem do Programa

```
0010 X=INT (RND (0)*20)+1
0020 Y=INT (RND (0)*20)+1
0030 IF X>=Y THEN 50
0040 Z=X:X=Y:Y=Z
0050 N=N+1
0100 PRINT "ESTOU A PENSAR EM DOIS NÚMEROS
ENTRE 1 E 20"
0110 PRINT "A SOMA DE AMBOS É"; X+Y
0120 PRINT "A DIFERENÇA ENTRE AMBOS É"; X-Y
0130 PRINT "QUAIS SÃO OS NÚMEROS?"
0140 INPUT X1:INPUT Y1
0150 IF X1<>X THEN 200
0160 IF Y1<>Y THEN 200
0170 PRINT "CERTO!"
0180 R=R+1
0190 GOTO 290
```

```
0200 IF X1<>Y THEN 250
0210 Y1 = X THEN 170
0250 PRINT "É PENA — ESTÁ ERRADO"
0260 PRINT "OS NÚMEROS ERAM ";X;" E ";Y
0290 PRINT "ESTÁ A PONTUAR "; R*100/N:"%"
0300 PRINT :PRINT "VAMOS JOGAR OUTRA VEZ!":
GOTO 10
```

Um exemplo de execução

```
ESTOU A PENSAR EM DOIS NÚMEROS ENTRE 1 E 20
A SOMA DE AMBOS É 13
A DIFERENÇA ENTRE AMBOS É 5
QUAIS SÃO OS NÚMEROS?
8
5
É PENA — ESTÁ ERRADO
OS NÚMEROS ERAM 9 E 4
ESTÁ A PONTUAR 0%

VAMOS JOGAR OUTRA VEZ!
ESTOU A PENSAR EM DOIS NÚMEROS ENTRE 1 E 20
A SOMA DE AMBOS É 10
A DIFERENÇA ENTRE AMBOS É 2
QUAIS SÃO OS NÚMEROS?
6
4
CERTO!
ESTÁ A PONTUAR 50%

VAMOS JOGAR OUTRA VEZ?
ESTOU A PENSAR EM DOIS NÚMEROS ENTRE 1 E 20
A SOMA DE AMBOS É 20
A DIFERENÇA ENTRE AMBOS É 2
QUAIS SÃO OS NÚMEROS?
11
9
CERTO!
ESTÁ A PONTUAR 66.6666666%
```



```
VAMOS JOGAR OUTRA VEZ!
ESTOU A PENSAR EM DOIS NÚMEROS ENTRE 1 E 20
A SOMA DE AMBOS É 10
A DIFERENÇA ENTRE AMBOS É 6
```

II

JOGO DE EQUILÍBRIO (1,2 K)

Se bem que este jogo simples e agradável seja também apropriado a qualquer computador ou calculadora programável, já não é tão simples de jogar, sendo de esperar que os aprendizes percam facilmente o equilíbrio — aliás, tal como nas situações reais. As crianças, em particular, adoram este jogo.

Quando sentir que joga bem, acrescente um atraso de tempo no loop de reacção:

210 W = W-D:D = P:X NEXT M

A oscilação (W) é então afectada pelo atraso de reacção (D), e a reacção actual (P) é guardada para a próxima vez que o loop é executado. A reacção só tem portanto efeito com um movimento de atraso — o que bastará para lhe estragar o jogo...

A linha 130 é particularmente interessante. A função RND dá um número aleatório na gama 0 a 0,99999, e o efeito da elevação deste número ao quadrado consiste em levar a maior parte dos números a ficarem na metade inferior dessa gama, só alguns conseguindo atingir a parte superior desta. Multiplicando por 8 e subtraindo 4 leva-se a instrução a dar um número aleatório na gama − 3 a +3, o que permite obter uma oscilação para a esquerda ou para a direita.

Lista de variáveis

Q$	String de entrada
A	Número de movimentos pretendidos
T	Posição de tabulação
M	Movimento actual
W	Oscilação para a esquerda ou direita
P	Ajustamento do jogador

Listagem do programa

```
0010 PRINT TAB (10); "JOGO DE EQUILÍBRIO"
0020 PRINT TAB (10); "= = = = = = = = = = = ="
0030 PRINT: PRINT
0040 INPUT "PRETENDE INSTRUÇÕES?", Q$
0050 IF LEFT$(Q$,1) = "s" THEN GOSUB 9000
0060 PRINT "EM QUANTOS MOVIMENTOS PENSA
QUE CONSEGUE"
0070 INPUT "EQUILIBRAR O TACO DE BILHAR?", A
0080 REM — Oscilar o taco
0090 T=10: FOR X =1 TO 10: PRINT TAB (T); "*":
NEXT X
0100 REM — Loop ****************
0110 FOR M=1 TO A
0120 IF W<0 THEN W=W-.5
0130 IF W=0 THEN X=RND: W=INT (X*X*8-4)
0140 IF W>0 THEN W=W+.5
0150 T =T+W: REM — Aumentar a oscilação
0160 IF T<=0 THEN 500
0170 IF T>=20 THEN 500
0180 PRINT TAB (T); "*": TAB (30);
0190 INPUT P: P=P-5
0210 W=W-P: NEXT M
0220 PRINT: PRINT "PARABÉNS — CONSEGUIU!"
0230 PRINT: INPUT "Joga outra vez (s/n)?", Q$
0240 IF Q$="s" THEN 60
0250 END
0500 PRINT:PRINT "OOOH! DEIXOU-O CAIR!": GO
TO 230
9000 PRINT "NESTE JOGO TEM DE EQUILIBRAR UM
TACO DE BILHAR";
9010 PRINT "NA TESTA. SE COMEÇAR A OSCILAR
PARA A ESQUERDA";
9020 PRINT "DEVE TAMBÉM IR PARA A ESQUERDA
A FIM DE COLOCAR-SE";
9030 PRINT "SOB O SEU CENTRO DE GRAVIDADE. SE
FOR PARA A";
9040 PRINT "DIREITA, DEVE IR PARA A DIREITA."
9050 PRINT "USE AS TECLAS DE NÚMEROS. O 5 É A
```

```
CENTRAL";
  9060 PRINT "1-4 PARA A ESQUERDA E 6-9 PARA A DI
REITA."
  9080 INPUT " CARREGUE EM RETURN.....", Q$
  9090 RETURN
```

Exemplo de execução

```
  PRETENDE INSTRUÇÕES? n
  EM QUANTOS MOVIMENTOS PENSA QUE CONSEGUE
  EQUILIBRAR O TACO DE BILHAR? 30
  PARABÉNS — CONSEGUIU!
  JOGA OUTRA VEZ (s/n)? n
  READY
```


III

JOGO DE MOEDAS (2 K)

São incluídas instruções na listagem, mas estas podem ser omitidas se assim se quiser ou se o programa não couber na máquina. O mesmo se aplica a qualquer outro programa apresentado neste livro — se for necessário poupar porque o tamanho da memória é reduzido, retire antes do mais as instruções (pode mantê-las num pequeno caderno de notas, ou consultar este volume).

Se ainda não consegue fazer caber o programa na máquina, retire-lhe todas as instruções REM, que apenas servem para tornar a listagem mais fácil de compreender. O computador não necessita delas. Se se habituar a fazê-lo, no entanto, deve ter cuidado; verifique se não existe qualquer outra linha onde se encontre uma instrução que envie para a linha que deseja retirar. Tentei evitar saltos para as linhas com instruções REM, mas dada a natureza humana, talvez tenha esquecido alguma.

Outro modo de poupar memória consiste em utilizar linhas de instruções múltiplas, se a máquina o permitir. Por exemplo, as linhas 100 e 110 podem ser juntas numa só, como de resto acontece com as linhas 120 e 130:

```
  100 PRINT : PRINT "POR FAVOR ESCREVA OS NO-
MES": P=1
  120 INPUT P$(P): IF P$(P)="" THEN 150
```

Não juntei linhas na listagem porque torna esta mais difícil de seguir.

A linha 40 contém um truque muito útil. Em resposta à linha 30, o utilizador pode escrever "NÃO" ou "SIM", ou apenas "N" ou "S". A linha 40 verifica apenas a primeira letra da entrada e actua em função dela, pelo que mesmo "SIM, POR FAVOR" será aceite. O programa não é parado por uma entrada

deste tipo, podendo o utilizador não se preocupar com o assunto.

A linha 130 evita a necessidade de dar entrada ao número de jogadores previamente — os nomes são simplesmente escritos com a extensão que se quiser e, quando tiverem sido completamente escritos, um novo toque na tecla RETURN faz a execução avançar.

Lista de variáveis

Q$	Entrada de respostas
P	Número total de jogadores
P$(X)	Nomes dos jogadores
C(X)	"Array" contendo a quantia pertencente a cada jogador
P(X)	"Array" contendo a aposta de cada jogador
X	Variável geral de contagem
C	Total de dinheiro
H	Número de caras
T	Número de coroas

Listagem do programa

```
0010 PRINT “JOGO DE MOEDAS”
0020 PRINT “----------------------------”
0030 INPUT “DESEJA INSTRUÇÕES?”, Q$
0040 IF LEFT$(Q$,1) = “N” THEN 100
0050 PRINT “DE CADA VEZ, O COMPUTADOR LAN
ÇA 16 MOEDAS.”
0060 PRINT “OS JOGADORES APOSTAM NO NÚMERO
DE CARAS QUE”
0070 PRINT “SAIRÃO. CADA UM APOSTA $1 E A PES
SOA QUE”
0080 PRINT “ACERTAR RECEBE TUDO. SE NINGUÉM
ACERTA, O”
0090 PRINT “DINHEIRO PASSA PARA A VEZ SEGUIN
TE.”
0100 PRINT:PRINT “POR FAVOR ESCREVA OS NO
MES”
```



```
0110 P = 1
0120 INPUT P$(P)
0130 IF P$(P) = “” THEN 150
0140 C(P) = 20:P = P + 1:GOTO 120
0150 PRINT “CADA JOGADOR COMEÇA COM 20 $”
0160 P = P-1
0170 FOR X = 1 TO P
0180 PRINT P$(X);:INPUT “— A SUA ADIVINHA”,
P(X)
0190 IF X = 1 THEN 240
0200 FOR Y = 1 TO X-1
0210 IF P(X)<>P(Y) THEN 230
0220 PRINT “ESSA APOSTA JÁ FOI FEITA”: GOTO 180
0230 NEXT Y
0240 (X) = C(X)−1:C = C = 1:NEXT X
0250 PRINT
0260 PRINT “VAMOS LANÇAR...”
0270 PRINT :H = 0:T = 0
0280 FOR X = 1 TO 16
0290 IF RND(0) . 5 THEN 310
0300 H = H + 1:PRINT “Ca”;:GOTO 320
0310 T = T + 1:PRINT “Co”;
0320 NEXT X
0330 REM — Verificar apostas
0340 FOR X = 1 TO P
0350 IF P(X)<>H THEN 400
0355 PRINT :PRINT
0360 PRINT P$(X); “GANHA! POSSUI AGORA”;
0370 C(X) = C(X) + C:C = 0
0380 PRINT “$”; C(X)
0390 GOTO 420
0400 NEXT X:PRINT
0410 PRINT H; “CARAS — NINGUÉM GANHA. O DI
NHEIRO NA MESA”
0415 PRINT “É AGORA $”;C
0420 PRINT “DINHEIRO RESTANTE — ”
0430 FOR X = 1 TO P
0440 PRINT P$(X): “ — ”:C(X)
0450 NEXT M
0460 FOR X = 1 TO P
```

```
0470 IF C(X) O THEN 500
0480 PRINT "TERMINOU O JOGO. OS JOGADORES
NÃO TÊM DINHEIRO"
0490 END
0500 NEXT X:GOTO 170
```

Exemplo de execução

```
Deseja instruções? N
Por favor escreva os nomes
? TOMÁS
? JOÃO
? JOSÉ
CADA JOGADOR COMEÇA COM 20 $
TOMÁS — A SUA ADIVINHA? 8
JOÃO — A SUA ADIVINHA? 7
JOSÉ — A SUA ADIVINHA? 6
VAMOS LANÇAR...
Ca Ca Ca Ca Co Ca Co Co Co Ca Ca Ca Ca Ca Co Co
10 CARAS — NINGUÉM GANHA. O DINHEIRO NA MESA
É AGORA $3
DINHEIRO RESTANTE —
TOMÁS — 19
JOÃO — 19
JOSÉ — 19
TOMÁS — A SUA ADIVINHA? 9
JOÃO — A SUA ADIVINHA? 8
JOSÉ — A SUA ADIVINHA? 6
VAMOS LANÇAR...
Co Ca Ca Ca Co Co Ca Co Co Co Ca Ca Ca Ca Ca Co
TOMÁS GANHA! POSSUI AGORA $24
DINHEIRO RESTANTE —
TOMÁS — 24
JOÃO — 18
JOSÉ — 18
TOMÁS — A SUA ADIVINHA? 7
JOÃO — A SUA ADIVINHA? 8
JOSÉ — A SUA ADIVINHA? 6
VAMOS LANÇAR...
```


IV

O SR. SILVA DIZ (3,5 K)

Neste jogo, o computador imprime um array. Mantém-no no visor durante alguns segundos e em seguida limpa-o antes de lhe pedir que repita a sequência. É portanto necessário ler o array tendo isto em conta.

O jogo requer evidentemente que este array tenha uma dimensão razoável. Como esta BASIC definirá normalmente um array até 10 elementos, e os utilizadores excederão facilmente isto ao testarem a sua memória, dimensionamos na linha 120 o array como tendo 30 elementos.

A linha 220 torna-se necessária para limpar o visor usando esta BASIC; os utilizadores usarão o correspondente nos seus sistemas.

Lista de variáveis

D	Grau de dificuldade; comprimento do array no princípio
D1	Grau de dificuldade; maior digito usado
X	contador geral
A$(X)	Array a memorizar
A	Entrada digital do utilizador
H	Maior pontuação até ao momento

Listagem do programa

```
0080 DIM A$(25), B$(25)
0090 PRINT CHR$(25); CHR$(25); CHR$(0); CHR$(0);
```

```
0100 PRINT TAB (20); "O SR. SILVA DIZ"
0110 PRINT TAB (20); "= = = = = = = ="
0120 PRINT
0130 INPUT "GRAU DE DIFICULDADE? ESCREVA UM
NÚMERO DE 0-9", D
0140 PRINT
0160 D=D+4
0170 R=INT (RND(0)*D+1)
0180 A$=A$+CHR$(R+48)
0190 FOR X=1 TO LEN(A$)
0195 PRINT MID$(A$,X,1);" ";
0196 FOR T=1 TO 50: NEXT T
0197 NEXT X
0200 FOR X=1 TO 100
0210 NEXT X
0220 PRINT CHR$(25); CHR$(22); CHR$(0)
0230 INPUT "REPITA A SEQUÊNCIA", B$
0240 IF A$-B$ THEN PRINT "CERTO!":GOTO 170
0250 PRINT: PRINT
0260 PRINT "RECORDOU";:L=LEN(A$)-1
0270 PRINT L; "DIGITOS CORRECTAMENTE, O QUE É"
0280 IF L>L1 THEN 300
0290 IF L=L1 THEN 310
0295 PRINT "MUITO BOM!": GOTO 320
0300 PRINT "A MELHOR PONTUAÇÃO DE HO
JE!!!":L1=L:GOTO 320
0310 PRINT "IGUAL À MELHOR PONTUAÇÃO ANTE
RIOR!":GOTO 320
0320 PRINT : PRINT:PRINT
0330 INPUT "OUTRO JOGO", A$
0340 IF LEFT$(A$,1)="N" THEN END
0350 A$=""
0360 GOTO 120
```

Exemplo de execução

```
GRAU DE DIFICULDADE? ESCREVA UM NÚMERO
DE 0-9? 0
 4 1 3 2
```



```
POR FAVOR REPITA A SEQUÊNCIA
4 /1 /3 /2 /CERTO!
 4 1 3 2 4
POR FAVOR REPITA A SEQUÊNCIA
4 /1 /3 /2 /4 /CERTO!
 4 1 3 2 4 1
POR FAVOR REPITA A SEQUÊNCIA
4 /1 /3 /2 /4 /1 /CERTO!
 4 1 3 2 4 1 3
POR FAVOR REPITA A SEQUÊNCIA
4 /1 /3 /2 /4 /1 /3 /CERTO!
 4 1 3 2 4 1 3 2
POR FAVOR REPITA A SEQUÊNCIA
4 /1 /3 /2 /4 /1 /3 /2 /CERTO!
 4 1 3 2 4 1 3 2 2
POR FAVOR REPITA A SEQUÊNCIA
4 /1 /3 /2 /4 /1 /3 /2 /2 /CERTO!
 4 1 3 2 4 1 3 2 2 1
POR FAVOR REPITA A SEQUÊNCIA
4 /1 /3 /2 /4 /1 /3 /2 /2 /1 /CERTO!
 4 1 3 2 4 1 3 2 2 1 1
POR FAVOR REPITA A SEQUÊNCIA
4 /1 /3 /2 /4 /1 /3 /2 /2 /1 /1 /CERTO!
 4 1 3 2 4 1 3 2 2 1 1 4
POR FAVOR REPITA A SEQUÊNCIA
4 /1 /3 /2 /4 /1 /3 /2 /2 /1 /1 /4 /CERTO!
 4 1 3 2 4 1 3 2 2 1 1 4 3
POR FAVOR REPITA A SEQUÊNCIA
4 /1 /3 /2 /4 /1 /3 /2 /2 /1 /1 /3 ***** ERRO *****
RECORDOU 12 DIGITOS CORRECTAMENTE, O QUE É
A MELHOR PONTUAÇÃO DE HOJE!!!
OUTRO JOGO?
N
READY
```

V

O JOGO DE KIM (2K)

Este jogo baseia-se livremente no livro de Rudyard Kipling. Você dispõe de trinta segundos para memorizar um conjunto de caracteres que (como em "O Sr. Silva diz") são impressos no visor do computador. Não interessa se os repete ou não pela ordem correcta, mas como é óbvio não pode indicar o mesmo mais do que uma vez a menos que tenha aparecido também mais de uma vez.

A diferença essencial entre "O Sr. Silva diz" e o "Jogo de Kim" é que, neste último caso, o utilizador encontra um maior número de elementos para memorizar de cada vez, dispondo por outro lado de mais tempo para o fazer. A qualidade do jogador é também diferente; enquanto "O Sr. Silva diz" não necessita de uma mente muito intelectual, o "Jogo de Kim" é mais adulto, encorajando o utilizador a classificar os elementos e a juntá-los mentalmente em grupos. Portanto, se bem que os dois jogos sejam superficialmente semelhantes, requerem métodos completamente diferentes de resposta da parte do jogador.

Nesta versão são usados até 60 elementos; este grande número pode ser usado por jogadores peritos devido à possibilidade de agrupamento dos elementos já mencionada — o jogador junta-os em digitos, letras maiúsculas, letras minúsculas, etc., talvez ainda subdividindo-as. Ocorrem necessariamente duplicações de elementos, tornando a solução um pouco mais simples.

O coração do programa encontra-se nas linhas 220 a 230. São produzidos números aleatórios indicados sob a forma de caracteres ASCII, e portanto passíveis de serem apresentados no visor. O valor mais pequeno é 33, correspondendo ao caracter "!", e o maior é 126, equivalente a "~", incluindo entre ambos todos os outros caracteres. Se os leitores não estão familiarizados com este código ASCII, podem adquirir uma obra que o apresente e estudá-lo.

Certos teclados muito usados não possuem teclas correspondentes a todos os símbolos e como o utilizador deve poder escrever neste jogo qualquer dos símbolos que vê no visor não pode permitir ao computador que imprima esses símbolos que não pode escrever em resposta. Este problema é resolvido nas linhas 226 e 227, que omitem o apóstrofo invertido (') e o sinal comercial @ respectivamente. Omita estas linhas, altere-se ou acrescente-as conforme for necessário para a sua máquina.

A linha 260 serve simplesmente para perder tempo. Obter o seno de um número é uma das funções mais lentas realizada pelo computador, dado que requer o uso de valores logaritmicos. Se a BASIC do seu computador não autoriza funções trigonométricas, altere simplesmente esta linha de modo a que o computador conte até um número muito grande, determinado por experiências.

A linha 280 é um modo alternativo de limpar o visor: imprimindo 16 linhas em branco. Se a sua máquina possui uma instrução CLS, use-a. Na linha 290 é permitido ao utilizador indicar um número de elementos superior em 25% aos elementos de facto apresentados para memorização.

Os caracteres apresentados foram produzidos e armazenados como valores numéricos, mas o utilizador é agora forçado a responder com os caracteres propriamente ditos; a ordem para substituir os caracteres pelo seu valor ASCII equivalente é ASC, que se pode ver na linha 320. As linhas que se seguem comparam cada valor ASCII com cada elemento do array, permitindo a escrita dos caracteres por qualquer ordem. Ao serem comparados, os elementos correspondentes do array são passados ao valor zero pela linha 340, o que obriga a escrever duas vezes os caracteres duplicados.

Finalmente, na linha 410 o computador imprime os caracteres que faltam, percorrendo muito simplesmente o array e imprimindo todos os elementos que não têm um valor zero.

Lista de variáveis

X(60) Array até 60 elementos

N	Número de elementos
X	Contador
Y	Contador
G	Número de adivinhas
R	Número de respostas correctas
W	Número de respostas erradas

Listagem do programa

```
0005 LINE= 32
0010 PRINT CHR$(25); xCHR$(25); CHR$(5); CHR$(14)
0020 DIM X(60)
0030 PRINT "ESTE É O JOGO DE KIM, ADAPTADO DO
LIVRO DE"
0040 PRINT "KIPLING. VOCÊ TEM 30 SEGUNDOS PA
RA MEMORIZAR"
0050 PRINT "UM CONJUNTO DE CARACTERES. PO
DE REPETI-LOS"
0060 PRINT "POR QUALQUER ORDEM, MAS DEVE
APRESENTÁ-LOS"
0070 PRINT "APENAS O NÚMERO DE VEZES QUE
APARECERAM"
0080 PRINT "ORIGINALMENTE"
0090 PRINT
0100 INPUT "QUANTOS CARACTERES QUER EXPERI
MENTAR", N
0110 IF N<=60 THEN 130
0120 PRINT "NÃO SEJA TÃO AMBICIOSO. O MÁXIMO
É 60": GOTO 100
0130 ON INT(N/10+1)GOTO 140, 150, 160, 210, 210, 190,
200
0140 PRINT "ENTÃO! VIVA PERIGOSAMENTE! TEN
TE 10 OU MAIS!": GOTO 100
0150 PRINT "HMMM — TEM MEDO, EH?": GOTO 210
0190 PRINT "AVENTUROSO!":GOTO 210
0200 PRINT "PERITO, EH?"
0210 PRINT:PRINT "EIS OS SEUS CARACTERES:"
0215 PRINT
0220 FOR Y =1 TO N
0225 X(Y)=INT(RND*93+33)
0226 IF X(Y)=96 THEN 225
0227 IF X(Y)=92 THEN 225
0230 PRINT CHR$(X(Y));" ";:NEXT Y: PRINT
0240 REM — Temporização
0245 PRINT
0250 PRINT "TEM 30 SEGUNDOS..."
0260 FOR X=1 TO 90:Y=SIN(X):NEXT X
0270 REM — Limpar visor e comparar
0280 PRINT CHR$(25); CHR$(25); CHR$(5); CHR$(14)
0290 PRINT "TEM"; INT(N+N/4); "HIPÓTESES. PA
RA TERMINAR, PRIMA RETURN";
0295 PRINT "DUAS VEZES. INDIQUE OS CARACTE
RES."
0300 G=0:R=0:W=0
0310 INPUT T$:IF T$="" THEN 360
0320 FOR Y=1 TO N:IF ASC(T$)=X(Y) THEN 340
0330 NEXT Y:W=W+1:GOTO 350
0340 X(Y)=0:R=R+1
0350 G=G+1:IF G<=INT(N+N/4)THEN 310
0360 PRINT :PRINT "TEM ";R;"CERTOS E "; W; "ER
RADOS EM ";G;"HIPÓTESES."
0380 IF W<>0 THEN 390
0385 IFR=N THEN 430
0390 PRINT :PRINT "FALHOU";
0400 FOR Y=1 TO N:IF X(Y)=0 THEN 420
0410 PRINT CHR$(X(Y));" ";
0420 NEXT Y:PRINT
0430 PRINT "A SUA PONTUAÇÃO É ";(R-W)/N*100
0440 INPUT "JOGA DE NOVO ",T$
0450 IF LEFT$(T$,1)="S" THEN 90
0460 PRINT  :PRINT "ADEUS. ESPERO QUE TENHA
GOSTADO."
0470 END
```

Exemplo de execução

```
QUANTOS CARACTERES QUER EXPERIMENTAR? 5
ENTÃO! VIVA PERIGOSAMENTE! TENTE 10 OU MAIS!
QUANTOS CARACTERES QUER EXPERIMENTAR? 15
HMMM — TEM MEDO, EH?
EIS OS SEUS CARACTERES:
M s 4 _ F9 ' 6 k V a Y m g 2
TEM 30 SEGUNDOS...
TEM 18 HIPÓTESES. PARA TERMINAR, PRIMA RETURN
DUAS VEZES. INDIQUE OS CARACTERES
? 2
? 4
? 6
? a
? m
? g
? s
? F
? M
? =
? )
?
TEM 9 CERTOS E 2 ERRADOS EM 11 HIPÓTESES.
FALHOU _ 9 ' k VY
A SUA PONTUAÇÃO É 46,6666666
JOGA DE NOVO? N
ADEUS. ESPERO QUE TENHA GOSTADO.
READY
```


VI

ALGARISMOS SIGNIFICATIVOS (3,5 K)

Este programa agradará a qualquer professor ou pai de uma criança que tenha problemas em compreender o valor de cada algarismo em função do lugar onde se encontra. O exemplo de execução explica tudo, pelo que podemos passar imediatamente à listagem.

O exercício requer o armazenamento dos três dígitos de um dado número separadamente, cada um deles com o seu valor total, isto é, o 2 de 235 deve ter o valor 20. Pode-se produzir três algarismos separados e multiplicar dois deles por 10 ou por 100, ou — como se fez aqui — pode-se produzir um número de três algarismos e dividi-lo. Os leitores poderão experimentar qual o método mais económico.

A linha 90 produz o número entre 100 e 999. Este é dividido no algarismo das centenas, o algarismo das dezenas e o das unidades nas linhas 100 a 112, enquanto as linhas 116 a 120 verificam se não existem dois iguais.

O computador apresenta à criança vinte problemas deste tipo e indica em seguida a sua pontuação.

Lista de variáveis

Q$	String de entradas
P	Números de problemas
S	Pontuação
N	Número de 3 algarismos original
H	Centenas
T	Dezenas
U	Unidades

X Contador
D Algarismo pedido
A Resposta da criança

Listagem do programa

```
0010 INPUT "Deseja instruções",Q$
0020 Q$=LEFT$(Q$,1)
0030 IF Q$="Y" THEN 9000
0040 REM — Limpar visor
0050 PRINT CHR$(25); CHR$(25); CHR$(5)
0060 PRINT :PRINT:PRINT
0070 PRINT TAB(25); "H.T.U."
0080 P=0:S=0
0090 N=INT(RND(O)*899+100)
0095 PRINT CHR$(6);
0100 H=INT(N/100)
0110 T=INT((N-H*100)/10)
0112 U=N-H*100-T*10
0114 REM — Verificar se não há dois algarismos iguais
0116 IF H=T THEN 90
0118 IF H=U THEN 90
0120 IF T=U THEN 90
0125 PRINT CHR$(22);
0130 PRINT TAB(25);H;T;U
0140 REM — Alterar posição do cursor
0150 PRINT  CHR$(16):FOR X=1 TO 6:PRINT
CHR$(10);:NEXT X
0160 REM — Escolher um algarismo
0170 X=INT(RND(0)*3)+1
0175 PRINT CHR$(22);:REM — Apagar
0180 PRINT "Qual é o valor de ";
0190 IF X=1 THEN PRINT H;:D=H*100:GOTO 220
0200 IF X=2 THEN PRINT T;:D=T*10:GOTO 220
0210 PRINT U;:D=U
0220 INPUT A
0230 IF A=D THEN PRINT " / CERTO!":GOTO 300
0240 PRINT
0250 PRINT "É pena — vale ";D;CHR$(11);
```



```
0260 REM — Dar tempo para ler a resposta
0270 FOR y=1 TO 6:Z=SIN(Y):NEXT Y
0280 IF S=0 THEN 500
0290 S=S-1:GOTO 500
0300 S=S+1
0310 P=PP 1:IF P>19 THEN 400
0320 REM — Mudar cursor, outro número
0330 PRINT CHR$(16):PRINT:PRINT:PRINT:PRINT
0340 GOTO 90
0400 PRINT "A sua pontuação foi ";S;" Quer jogar outra vez";
0410 INPUT Q$
0420 IF LEFT$(Q$,1)="Y" THEN 50
0430 PRINT "Adeus. Obrigado por ter jogado.":END
0500 REM — Mover cursor, apagar e repetir a pergunta
0510 PRINT CHR$(11);CHR$(13);:GOTO 175
0999 END
9000 PRINT CHR$(25);CHR$(25);CHR$(14)
9010 REM — modo scrolling
9020 PRINT "Isto é muito fácil. Vou dar-lhe um número de ";
9030 PRINT "3 algarismos e perguntar quanto vale um destes. ";
9040 PRINT "Por exemplo —"
9050 PRINT :PRINT TAB (25); "3 4 5"
9060 PRINT :PRINT "Quanto vale o 3?"
9070 PRINT "Deve escrever depois a resposta, que neste caso";
9080 PRINT "é 300. Se lhe perguntar o valor do 5, a resposta";
9090 PRINT "será 5! O 4 vale 40. ";
9100 PRINT "Já sabia, não é verdade?"
9110 PRINT :PRINT "Carregue em RETURN quando esti
ver pronto";: INPUT A$
9120 GOTO 50
```

Exemplo de execução

Deseja instruções? Y

Isto é muito fácil. Vou dar-lhe um número de 3 algarismos e perguntar quanto vale um deles. Por exemplo —

3 4 5

Quanto vale o 3?

Deve escrever depois a resposta, que neste caso é 300. Se lhe

perguntar o valor de 5, a resposta será 5! O 4 vale 40.
Já sabia, não é verdade?
Carregue em RETURN quando estiver pronto

```
                                   H.T.U
                                   8 0 7
Quanto vale o 0? 0 / CERTO!
                                   H.T.U
                                   9 2 6
Quanto vale o 2? 2
É pena — vale 20
Quanto vale o 2? 20 / CERTO!
                                   H.T.U
                                   4 8 1
Quanto vale o 8? 80 / CERTO!
                                   H.T.U
                                   7 3 8
Quanto vale o 8? 80
É pena — vale 8
Quanto vale o 8? 8 / CERTO!
                                   H.T.U
                                   7 3 8
Quanto vale o 8? 8 / CERTO!
                                   H.T.U
                                   4 2 5
Quanto vale o 5? 5 / CERTO!
                                   H.T.U
                                   2 7 8
Quanto vale o 7? 70 / CERTO!
                                   H.T.U
                                   5 2 9
Quanto vale o 9? 9 / CERTO!
                                   H.T.U
                                   2 8 9
Quanto vale o 9? 9 / CERTO!
```


VII

FIZZ BUZZ (1,5 K)

Apresentamos aqui uma adaptação para computador de um jogo popular no qual se diz "fizz" em vez de qualquer múltiplo de três, "buzz" em vez de um múltiplo de 5 e "fizz buzz" em vez de um número divisível por 3 e por 5. Não necessitamos de um array, dado que a sequência de números e respostas é fácil de obter aritmeticamente.

A listagem introduz a ideia de uma "flag" — bandeira — que tal como a de um guarda de passagem de nível é levantada quando estão reunidas certas condições. Trata-se de um conceito muito importante em programação, e o leitor virá a encontrar muitos exemplos ainda neste livro. Como é óbvio, as condições a ter em conta aqui são a possibilidade de cada novo número da série ser divisível por (a) 3, (b) 5, ou (c) ambos. Em todos os outros casos a "flag" não será "levantada". Ao contrário da bandeira do guarda, no entanto, a nossa terá quatro estados possíveis; é fácil definir cada um destes dando-lhe um valor numérico entre 0 e 3, onde 0 corresponde a estar "baixada".

A série de números começa na linha 60 e é incrementada de cada vez que a execução passa pela linha 100. A linha 110 verifica se S é divisível por 3. A variável S é primeiramente dividida por 3, arredondada (pela função INTEGER) e em seguida de novo multiplicada por 3. Como é óbvio, se o resultado tiver o mesmo valor que S, este será um múltiplo de 3 e a bandeira é levantada. O mesmo acontece na linha seguinte, pelo que a bandeira é levantada (ou não) antes de se pedir uma entrada ao utilizador.

Para compreender o que se segue, é necessário saber que o valor ou código ASCII de "0" é 48, enquanto que o do 9 é 57. As letras maiúsculas têm códigos entre 65 e 90, inclusivé, pelo que no caso de o computador detectar a entrada de uma palavra

(linha 140) salta para a linha 200. Se o código ASCII da entrada se encontra entre 58 e 64 inclusivé terá sido cometido um erro, tal como no caso de o valor ser inferior a 48. Veja as linhas 150 a 170.

A linha 180 toma o valor da string de entrada (um número) e atribui-o à variável V para comparação ulterior. Em seguida verifica-se o estado da flag (linha 185).

A flag é igualmente observada na linha 200, sendo realizado um salto condicionado em função do seu valor, de modo a comparar as palavras na entrada com o que deveriam ser.

Lista de variáveis

H	Maior pontuação actual
S	Número
F	Flag
Q$	Entradas
V	Valor numérico de Q$

Listagem do programa

```
0010 PRINT TAB (20); "FIZZ-BUZZ"
0020 PRINT TAB (20); "= = = = = ="
0030 H=0
0040 PRINT :PRINT "Deseja instruções?"
0050 INPUT Q$:IF LEFT$(Q$,1)="Y" GOSUB 9000
0060 S=0
0100 S=S+1:F=0:REM — Baixar Flag
0110 IF INT(S/3)*3=S THEN F=1
0120 IF INT(S/5)*5=S DHEN F=F+2
0130 INPUT "Indique um número", Q$
0140 IF ASC(Q$)>64 THEN 200
0150 IF ASC(Q$)< 58 THEN 170
0160 PRINT "Entrada incorrecta!";GOTO 130
0170 IF ASC(Q$)<48 THEN 160
0180 V=VAL(Q$)
0185 IF F<>0 THEN 500
0190 IF V=S PRINT "Certo!":GOTO 100
```



```
0195 GOTO 500
0200 ON F+1 GOTO 500, 210, 250, 300
0210 IF Q$="FIZZ" PRINT "Certo!":GOTO 100
0220 GOTO 500
0250 IF Q$="BUZZ" PRINT "Certo!":GOTO 100
0260 GOTO 500
0300 IF Q$="FIZZ BUZZ" PRINT "Certo!":GOTO 100
0500 PRINT :PRINT
0510 PRINT "É pena — devia ter sido ";
0520 ON F+1 GOTO 530, 550, 600, 650
0530 PRINT S:GOTO 700
0550 PRINT "FIZZ":GOTO 700
0600 PRINT "BUZZ":GOTO 700
0650 PRINT "FIZZ BUZZ"
0700 PRINT :PRINT "A sua pontuação era ";S-1;" o que é ";
0710 PRINT :PRINT "bom.":GOTO 750
0720 PRINT "a melhor de hoje!"
0730 H=S-1
0750 PRINT :PRINT "Jogo seguinte — ":GOTO 40
9000 PRINT "Escreva os números um de cada vez, começan
do";
9005 PRINT "pelo menor. ";
9010 PRINT "Não se esqueça de carregar em RETURN de
pois ";
9020 PRINT "de cada um. Mas não escreva os divisíveis
por ";
9030 PRINT "3. Em vez disso, escreva FIZZ. Do mesmo mo
do, ";
9040 PRINT "escreva BUZZ em vez dos divisíveis por 5.
Quando ";
9050 PRINT "o número é divisível por 3 e por 5, escreva ";
9060 PRINT "FIZZ BUZZ. Carregue em RETURN."
9100 INPUT Q$:RETURN
```

Exemplo de execução

```
Deseja instruções? N
Indique um número? 1
Certo!
```

Indique um número? 2
Certo!
Indique um número? 3
É pena — devia ter sido FIZZ
A sua pontuação era 2 o que é a melhor de hoje!
Jogo sèguinte —
Deseja instruções? N
Indique um número? 1
Certo!
Indique um número? 2
Certo!
Indique um número? FIZZ
Certo!
Indique um número? 4
Certo!
Indique um número? BUZZ
Certo!
Indique um número? FIZZ
Certo!
Indique um número? 7
Certo!
Indique um número? 8
Certo!
Indique um número? FIZZ
Certo!
Indique um número? BUZZ
Certo!
Indique um número? 11
Certo!
Indique um número? FIZZ
Certo!
Indique um número? 13
Certo!
Indique um número? 14
Certo!
Indique um número? FIZZ BUZZ
Certo!
Indique um número? 16
Certo!
Indique um número? 17

Certo!
Indique um número? FIZZ
Certo!
Indique um número? 19
Certo!
Indique um número? BUZZ
Certo!
Indique um número? 21
É pena — devia ter sido FIZZ
A sua pontuação é 20 o que é a melhor de hoje!
Jogo seguinte —
Deseja instruções? N

VIII

DISCURSOS (2,5 K)

Não há muito para dizer àcerca deste programa bastante simples. O objectivo consiste em utilizar três listas de palavras, escolher ao acaso uma palavra de cada uma delas e juntar as três numa frase (uma string). A ideia parece óptima, mas os resultados muitas vezes não têm sentido. Este programa em particular pôs vários professores a rir, pois obviamente as palavras foram escolhidas tendo em vista esta profissão. O método permite muitas adaptações; bastará escolher as palavras apropriadas.

Não consigo deixar de pensar que os nossos políticos costumam usar um programa deste género para fazerem os seus discursos...

A linha 25 define a largura da linha que é impressa no visor. Omita-a se quiser.

Lista de variáveis

D	Número de frases desejado
E	Actual número de frase
A	Número escolhido ao acaso

Listagem do programa

```
0010 PRINT "DISCURSOS"
0020 PRINT
0025 LINE = 56
0030 PRINT "ESTE PROGRAMA IMPRIME FRASES
MUITO ACEITÁVEIS ";
```



```
0050 PRINT "QUE PODEM SER APROVEITADAS EM
DISCURSOS. ";
0055 PRINT "PODEM-SE USAR ATÉ 14000."
0060 PRINT
0070 PRINT "QUANDO SURGIR UM '?', ESCREVA O
NÚMERO DE ";
0080 PRINT "FRASES DESEJADO. ESCREVA. '0' PARA
TERMINAR."
0100 PRINT
0110 INPUT D:IFD = 0 THEN END
0115 FOR E = 1 TO D
0120 A = INT(RND(0)*30) + 1
0140 IF A = 1 PRINT "AGRUPAMENTO"
0145 IF A = 2 PRINT "MODIFICAÇÃO"
0150 IF A = 3 PRINT "AMBIENTE"
0155 IF A = 4 PRINT "RESPONSABILIDADE"
0160 IF A = 5 PRINT "PROCESSO"
0165 IF A = 6 PRINT "CURRICULUM"
0170 IF A = 7 PRINT "ALGORITMO"
0175 IF A = 8 PRINT "RESULTADO"
0180 IF A = 9 PRINT "REFORÇO"
0185 IF A = 10 PRINT "AULA"
0190 IF A = 11 PRINT "RECURSO"
0200 IF A = 12 PRINT "ESTRUTURA"
0205 IF A = 13 PRINT "INSTALAÇÃO"
0210 IF A = 14 PRINT "SITUAÇÃO"
0215 IF A = 15 PRINT "APRENDIZADO"
0220 IF A = 16 PRINT "COMPORTAMENTO'
0225 IF A = 17 PRINT "AVALIAÇÃO"
0230 IF A = 18 PRINT "RELATÓRIO"
0235 IF A = 19 PRINT "IMPACTO"
0240 IF A = 20 PRINT "DESENVOLVIMENTO"
0250 IF A = 21 PRINT "DISSEMINAÇÃO"
0255 IF A = 22 PRINT "RECOMENDAÇÃO"
0260 IF A = 23 PRINT "AUSÊNCIA"
0265 IF A = 24 PRINT "CONGRESSO"
0270 IF A = 25 PRINT "PARTICIPAÇÃO"
0280 IF A = 26 PRINT "ÍMPETO"
0285 IF A = 27 PRINT "TÉCNICA"
0290 IF A = 28 PRINT "EXPERIÊNCA"
```

```
0295 IF A=29 PRINT "CONSÓRCIO"
0300 IF A=30 PRINT "DATA"
1000 PRINT " ";
1010 A=INT(RND(0)*21)+1
1020 IF A=1 PRINT "HUMANÍSTICO"
1025 IF A=2 PRINT "SEQUENCIAL"
1030 IF A=3 PRINT "CLASSIFICÁVEL"
1035 IF A=4 PRINT "SUBSTANTIVO"
1040 IF A=5 PRINT "NOMINAL"
1045 IF A=6 PRINT "INTEGRADO"
1050 IF A=7 PRINT "CRIATIVO"
1055 IF A=8 PRINT "MOTIVACIONAL"
1060 IF A=9 PRINT "VERTICAL"
1065 IF A=10 PRINT "PSICOLÓGICO"
1070 IF A=11 PRINT "AUTORITÁRIO"
1080 IF A=12 PRINT "OBJECTIVO"
1085 IF A=13 PRINT "EXTRA-ASSOCIATIVO"
1090 IF A=14 PRINT "CONSULTIVO"
1100 IF A=15 PRINT "NORMATIVO"
1105 IF A=16 PRINT "COGNITIVO"
1110 IF A=17 PRINT "PROJECTIVO"
1115 IF A=18 PRINT "ANALÍTICO"
1120 IF A=19 PRINT "PROGRAMÁVEL"
1125 IF A=20 PRINT "NORMAL"
1130 IF A=21 PRINT "COMPOSTO"
1500 PRINT " ";
1505 A=INT (RND(0)*23)+1
1510 IF A=1 PRINT "BÁSICO"
1515 IF A=2 PRINT "COMPORTAMENTISTA"
1520 IF A=3 PRINT "ORIENTADOR"
1525 IF A=4 PRINT "FLEXÍVEL"
1530 IF A=5 PRINT "DIFERENCIADO"
1535 IF A=6 PRINT "DESCOBERTO"
1540 IF A=7 PRINT "COMENSURÁVEL"
1545 IF A=8 PRINT "HOMOGÉNEO"
1550 IF A=9 PRINT "HETEROGÉNEO"
1555 IF A=10 PRINT "MANIPULADOR"
1560 IF A=11 PRINT "MODULAR"
1565 IF A=12 PRINT "PEDAGÓGICO"
1570 IF A=13 PRINT "APROPRIADO"
```



```
1575 IF A=14 PRINT "INTRÍNSECO"
1580 IF A=15 PRINT "ILIMITADO"
1585 IF A=16 PRINT "PRIORITÁRIO"
1590 IF A=17 PRINT "INFORMATIVO"
1600 IF A=18 PRINT "IMPLEMENTÁVEL"
1605 IF A=19 PRINT "CORRECTO"
1610 IF A=20 PRINT "AVANÇADO"
1615 IF A=21 PRINT "POTENCIAL"
1620 IF A=22 PRINT "PARTICIPATIVO"
1625 IF A=23 PRINT "APLICÁVEL"
2000 PRINT :NEXT E:GOTO 60
```

Exemplo de execução

```
ESTE PROGRAMA IMPRIME FRASES MUITO ACEITÁ-
VEIS QUE PODEM SER APROVEITADAS EM DISCUR-
SOS. PODEM-SE USAR ATÉ 14000.

QUANDO SURGIR UM '?', ESCREVA O NÚMERO DE
FRASES DESEJADO.
ESCREVA '0' PARA TERMINAR.

REFORÇO MOTIVACIONAL PEDAGÓGICO
INSTALAÇÃO CRIATIVO MODULAR
RESPONSABILIDADE PROJECTIVO AVANÇADO
DISSEMINAÇÃO COGNITIVO POTENCIAL
IMPACTO VERTICAL INTRÍNSECO
REFORÇO MOTIVACIONAL HETEROGÉNEO
CURRICULUM CONSULTIVO PARTICIPATIVO
```

IX

PUZZLE (2,5 K)

Muitos leitores já viram certamente pequenos puzzles em plástico constituídos por um grupo de letras que devem ser colocadas por uma dada ordem. Este jogo faz exactamente isso, pelo que não vale a pena perder tempo a descrevê-lo.

Os leitores que estão a aprender a programar devem notar que apesar de a área de jogo ser uma matriz de 4×4 unidades, o programa mostra que não é necessária uma matriz para o executar num computador. Um array servirá igualmente bem. Aliás, quando pensamos no assunto, verificamos que o conceito de uma matriz de computador não tem de facto sentido dado que o espaço de memória tem uma concepção linear. A BASIC permite-nos usar matrizes através de uma manipulação inteligente dos endereços das variáveis, mas daí decorre um custo em tempo de execução e em memória adicional que nem sempre se justifica em termos de conveniência de programação. Com efeito, o uso de um array em vez de uma matriz é praticamente equivalente na aplicação aqui considerada.

Um array B$ contém as letras do puzzle, e a linha 110 escolhe as letras A-P por uma ordem aleatória colocando-as numa zona de armazenamento temporário T$. As linhas que se seguem verificam se a letra já foi usada anteriormente, colocando-a em seguida no array. Nestas condições, a situação de partida é completamente aleatória.

As linhas entre 170 e 230 imprimem o array sob a forma de uma matriz, deixando um espaço em branco em vez do "P". Pede-se então ao utilizador que indique uma entrada, e a apresentação de um "X" neste momento provoca o fim do jogo. Em seguida o programa verifica se o caracter apresentado é legítimo (linha 260) antes de verificar se o movimento é legal (linhas

300-340). Depois de cada movimento verifica-se se as letras se encontram pela ordem correcta.

Se puder colocar o seu computador em modo "página" e desligar o cursor, as letras parecerão mover-se sozinhas e o jogo tornar-se-á mais rápido. Deverá trazer o cursor à origem na linha 160 se o quiser fazer. Trata-se de uma pequena modificação que merece ser introduzida, dado que observar o visor rolando pode tornar-se cansativo para os olhos ao fim de algum tempo.

Lista de variáveis

B$(X)	Array de letras misturadas
A$	String de resposta
M	Número de movimentos
T$	String de transferência temporária
X	Variável de contagem
S	Posição do espaço (ou "P") no array
I	Zona de armazenamento temporário para S; também para uso geral
J	Variável de contagem

Listagem do programa

```
0010 PRINT TAB(5);"PUZZLE"
0020 PRINT TAB(5); "------------"
0030 PRINT :PRINT
0040 DIM B$(16)
0050 INPUT "Deseja instruções", A$
0060 IF LEFT$(A$,1) = "N" THEN 95
0070 PRINT "ESTE PROGRAMA É UMA CÓPIA DOS
PEQUENOS PUZZLES ";
0072 PRINT "DE PALAVRAS QUE JÁ CONHECE. DE
VE DISPOR AS ";
0074 PRINT "LETRAS PELA ORDEM CORRECTA
MOVENDO-AS UMA DE ";
0076 PRINT "CADA VEZ. A POSIÇÃO FINAL É _"
0078 PRINT "     A B C D"
0080 PRINT "     E F G H"
```

```
0082 PRINT "      I J K L"
0084 PRINT "      M N O"
0086 PRINT "QUANDO CARREGAR EM RETURN, LE
VAREI CERCA DE ";
0088 PRINT "10 SEGUNDOS A CONSTRUIR O PUZZ
LE...".
0090 INPUT A$
0095 M=0:REM — Definir condições iniciais
0100 FOR I=1 TO 16
0105 PRINT CHR$(25);
0110 T$=CHR$(INT(RND*16+65))
0120 FOR J=1 TO I
0130 IF B$(J)=T$ THEN 110
0140 NEXT J
0150 B$(I)=T$
0155 IF T$="p" S=I
0160 NEXT I
0165 REM — Imprimir matriz
0170 X=1:FOR D=1 TO 3:PRINT:NEXT D
0180 FOR I=J TO 4:PRINT TAB(10);
0190 FOR J=1 TO 4
0210 IF B$(X)="P" THEN PRINT " "; GOTO 220
0215 PRINT B$(X);" ";
0220 X=X+1:NEXT J: PRINT
0230 NEXT I:PRINT
0240 INPUT "Letra a mover",A$
0245 IF A$="X" THEN 450
0250 FOR X=1 TO 16
0260 IF A$=B$(X) THEN 290
0270 NEXT X
0280 PRINT CHR$(11)
0285 PRINT "ENTRADA INCORRECTA — ";:GOTO 240
0290 REM — Verificar legalidade
0300 IF ABS(S-X)=4 THEN 360
0305 IF ABS(S-X)<>1 THEN 350
0310 IF X/4<>INT(X/4) THEN 330
0320 IF S=X+1 THEN 350
0330 IF (X-1)/4<>INT((X-1)/4) THEN 360
0340 IF S<>X-1 THEN 360
0350 PRINT CHR$(11);
```



```
0355 PRINT "MOVIMENTO ILEGAL — ";:GOTO 240
0360 REM — Mover
0365 M=M+1
0370 B$(X)="P"
0380 B$(S)=A$
0390 S=X
0392 FOR U=1 TO 9
0393 PRINT CHR$(11);:NEXT U
0395 REM — Verificar a solução
0400 FOR I=1 TO 16
0410 IF CHR$(I+64)<>B$(I) THEN 170
0420 NEXT I
0430 PRINT
0440 PRINT "Óptimo — resolveu o problema em "; M;"
movimentos."
0450 PRINT
0460 PRINT "Quer jogar novamente?"
0470 INPUT A$
0480 IF LEFT$(A$,1)="Y" THEN 95
0490 PRINT
0500 PRINT "Adeus — e obrigado pelo jogo."
```

Exemplo de execução

Deseja instruções? N

```
E A B H
I F L M
N D G O
J   C K
```

Letra a mover? D

```
E A B H
I F L M
N   G O
J D C K
```

Letra a mover? F

```
E A B H
I   L M
N F G O
J D C K
```

Letra a mover? I

```
E A B H
  I L M
N F G O
J D C K
```

Letra a mover? E

```
  A B H
E I L M
N F G O
J D C K
```

Letra a mover? A

```
A   B H
E I L M
N F G O
J D C K
```

Letra a mover? B

```
A B   H
F T I M
```


X

EXERCÍCIO DE ESCRITA (3 K)

Foram já feitas muitas tentativas para conceber um teste de escrita capaz de ser realizado por um computador, mas até agora o êxito foi limitado. Uma linha de experimentação bastante prometedora consiste em gravar uma voz que diz a palavra a escrever, mas o autor (e não só) verificou que não se pode confiar nas crianças para manipularem um gravador de fita e simultaneamente concentrarem-se na pronúncia das palavras. Mesmo o controlo do gravador pelo computador não mostrou ser inteiramente satisfatório devido ao facto de a fita e o computador deixarem facilmente de estar sincronizados; e o uso de equipamento caro ou pouco habitual está fora de causa.

Uma das dificuldades de qualquer teste impresso, em vez de falado, consiste na necessidade de evitar dar quaisquer pistas ao utilizador. Simultaneamente, torna-se necessário dar a este uma informação suficiente quanto à palavra que deve ser escrita.

É certo que muitas pessoas possuem uma memória visual que se torna muito útil ao escrever palavras; isto é, recordam a aparência da palavra. Se estas pessoas enfrentam um teste do tipo habitual, em que devem escolher uma entre cinco (por exemplo) ortografias alternativas, dispõem evidentemente de uma vantagem relativamente àqueles que não possuem essa memória visual.

É improvável que surja alguma solução verdadeiramente satisfatória para este problema até os computadores serem tão baratos que possamos ter vocabulários muito extensos armazenados na máquina mais ou menos permanentemente — e mesmo então haverá problemas no caso de dialectos regionais.

Uma outra solução possível consistiria na programação de texto, usando o método dos fonemas no qual o texto é recons-

truído em vez de propriamente guardado em memória, mas isto requer um hardware especial. Pior ainda, a eliminação de incompreensões provocadas pela voz um tanto metálica não é fácil de conseguir quando são ditas palavras isoladas em vez de frases completas.

O método usado neste programa, portanto, não é de modo algum perfeito, mas é talvez o melhor disponível com meios simples. São mostrados ao utilizador quatro modos diferentes de escrever uma mesma palavra, qualquer deles podendo estar correcto. Pode também acontecer que nenhum o esteja; esta última possibilidade é importante e dá um maior interesse ao teste. Permite ainda à pessoa conceber a lista de palavras de modo a lançar pistas erradas.

O programa é simples, mas ilustra pela primeira vez neste livro o uso de instruções DATA — linha 1000 até ao fim. O computador nada faz quando e se encontrar estas linhas, só as considerando quando encontra uma instrução READ como a da linha 130. A instrução READ A$ considera o elemento seguinte das instruções DATA, atribui-o à string A$, e simultaneamente altera um "ponteiro" (*pointer*) interno que passa a designar o elemento seguinte, que será lido pela próxima instrução READ.

Os utilizadores podem facilmente conceber os seus próprios testes de escrita alterando estas instruções DATA. Se o leitor resolver fazê-lo, note que cada conjunto de quatro palavras deve sempre conter uma quinta, que indica a escrita correcta ao computador. Esta palavra não é necessariamente apresentada ao utilizador, mas serve à máquina para verificar a correcção da resposta.

O teste é apresentado aqui na sua versão original, em inglês, não sendo difícil substituir as palavras por outras da nossa língua. Tal como está, porém, pode também ser útil a algum estudante de inglês...

Lista de variáveis

A$	String a transferir para o visor; também resposta do utilizador.
B$	String contendo a escrita correcta.
S	Pontuação.
X	Contagem do número de problemas apresentados.

Listagem do programa

```
0010 PRINT :PRINT TAB(15), "TESTE DE ESCRITA 1"
0020 PRINT TAB(15), "= = = = = = = = = = = ="
0030 PRINT :PRINT
0040 PRINT "Neste teste, apresento-lhe quatro ortografias"
0050 PRINT "de uma palavra. Só uma — ou nenhuma — estará"
0060 PRINT "correcta. Deve descobrir qual é."
0070 PRINT "No final indicarei a sua pontuação. Terá de"
0080 PRINT "resolver 20 problemas."
0090 S=0:PRINT:PRINT: PRINT "Carregue em RETURN
quando quiser começar."
0100 DIM A$(20),B$(20)
0110 INPUT A$
0120 FOR X=1 TO 20
0130 FOR Y=1 TO 4:READ A$
0140 PRINT A$;" ";:NEXT Y:PRINT
0150 READ B$
0160 INPUT "A SUA ORTOGRAFIA", A$
0170 IF A$=B$ THEN 200
0180 PRINT "ERRADO. A RESPOSTA CORRECTA É
";B$
0190 PRINT :PRINT:GOTO 230
0200 PRINT "CERTO."
0210 S=S+1
0220 PRINT :PRINT
0230 NEXT X
0240 PRINT :PRINT "TERMINOU O TESTE."
0250 PRINT "VOCÊ PONTUOU ";S; "NUM TOTAL DE
20"
0260 IF S<>20 THEN END
0270 FOR X=1 TO 20
0280 PRINT CHR$(30);CHR$(7);
0290 NEXT X
0300 PRINT "PARABÉNS! UMA PONTUAÇÃO ÓPTIMA!"
0310 END
1000 DATA PASENGER, PASSINGER, PASENJER, PAS-
SENGER, PASSENGER
1010 DATA APROCHE, APPROACH, APROACH,
APROACHE, APPROACH
```

```
1020 DATA CREDDIT, CREDIT, KREDDIT, CREDITT,
CREDIT
1030 DATA SHEPHERD, SHEEPHERD, SHEPPARD,
SHEPPERD, SHEPHERD
1040 DATA ENCURRIGE, ENCOUREGE, ENCOURAGE,
ENCURRAGE, ENCOURAGE
1050 DATA ELEXTION, ELECTION, ELECSHUN,
ELECSHON, ELECTION
1060 DATA ANDSOME, HANDSOME, HANSUM, HAN-
SUME, HANDSOME
1070 DATA RASCAL, RASCLE, RASKLE, RASCEL,
RASCAL
1080 DATA REZEMBLE, RESEMBEL, RESEMBLE, RES-
SEMBLE, RESEMBLE
1090 DATA WRESTEL, RESTLE, RESSEL, RESSAL,
WRESTLE
1100 DATA POZESS, POSSESS, POSESS, POSSES, POS-
SESS
1110 DATA ATENSHUN, ATTENSHON, ATENSHON,
ATENTION, ATTENTION
1120 DATA BERGLER, BERGLAR, BURGLER, BURGU-
LER, BURGLER
1130 DATA PLUNGE, PLUNJE, PLUNJE, PLUNDGE,
PLUNGE
1140 DATA GOVERMENT, GOVERNMENT, GOVVER-
MENT, GUVERMENT, GOVERNMENT
1150 DATA CRUELLY, CRUELY, CRUELEY, CREWE-
LY, CRUELLY
1160 DATA PURCHASE, PERCHASE, PERCHAISE,
PERCHASE, PURCHASE
1170 DATA LIDHTENING, LITENING, LIGHTNIN, LI-
TENIN, LIGHTNING
1180 DATA TUNG, TUNGUE, TONGUE, TUNGE, TON-
GUE
1190 DATA IMPROOVEMENT, IMPROVMENT, INPRO-
VEMENT, INPROOVEMENT, INPROVEMENT
```


Exemplo de execução

```
Neste teste, apresento-lhe quatro ortografias
de uma palavra. Só uma — ou nenhuma — estará
correcta. Deve descobrir qual é.
No final indicarei a sua pontuação. Terá de
resolver 20 problemas.

Carregue em RETURN quando quiser começar.
PASENGER PASSINGER PASENJER PASSENGER

A SUA ORTOGRAFIA? PASSENGER
CERTO.

APPOCHE APPROACH APROACH APROACHE
A SUA ORTOGRAFIA? APROACH
ERRADO. A RESPOSTA CORRECTA É APPROACH

CREDDIT CREDIT KREDDIT CREDDITT
A SUA ORTOGRAFIA? CREDIT
CERTO.

SHEPERD SHEEPHERD SHEPPARD SHEPPERD
A SUA ORTOGRAFIA? SHEPEARD
ERRADO. A RESPOSTA CORRECTA É SHEPHERD

ENCURRIGE  ENCOUREGE  ENCOURAGE  ENCURRAGE
A SUA ORTOGRAFIA? ENCOURAGE
CERTO.
ELEXTION ELECTION ELECSHUN ELESCHON
A SUA ORTOGRAFIA? ELECTION
CERTO.

ANDSOME HANDSOME HANDSUM HANDSUME
A SUA ORTOGRAFIA? HANDSOME
CERTO.

RASCAL RASCLE RASKLE RASCEL
A SUA ORTOGRAFIA? RASCAL
CERTO.
```

REZEMBLE RESEMBEL RESEMBLE RESSEMBLE
A SUA ORTOGRAFIA? RESEMBLE
CERTO.

WRESTEL RESTLE RESSEL RESSAL
A SUA ORTOGRAFIA? WRESTLE
CERTO.

POZESS POSSESS POSESS POSSES
A SUA ORTOGRAFIA? POSSESS
CERTO.

ATENSHUN ATTENSHON ATENSHON ATENTION
A SUA ORTOGRAFIA? ATENTION
ERRADO. A RESPOSTA CORRECTA É ATTENTION

BERGLER BERGLAR BURGLER BURGULER
A SUA ORTOGRAFIA? BUGLAR
ERRADO. A RESPOSTA CORRECTA É BURGLAR

PLUNGE PLUNJE PLUNJE PLUNDGE
A SUA ORTOGRAFIA? PLUNGE
CERTO.

GOVERMENT GOVERNMENT GOVVERMENT GUVER-MENT
A SUA ORTOGRAFIA? GOVERMENT
ERRADO. A RESPOSTA CORRECTA É GOVERNMENT

CRUELLY CRUELY CRUELEY CREWELY
A SUA ORTOGRAFIA? CRUELLY
CERTO.

PURCHASE PERCHASE PERCHAISE PERCHASE
A SUA ORTOGRAFIA? PURCHASE
CERTO.

LIGHTENING LITENING LIGHTNIN LITENIN
A SUA ORTOGRAFIA? LIGHTNING
CERTO.

TUNG TUNGUE TONGUE TUNGE
A SUA ORTOGRAFIA? TONGUE
CERTO.

IMPROOVEMENT IMPROVMENT INPROVEMENT IN-PROOVEMENT
A SUA ORTOGRAFIA? IMPROVEMENT
CERTO.

TERMINOU O TESTE.
VOCÊ PONTUOU 15 NUM TOTAL DE 20

XI

ARITMÉTICA (1,5 K)

Apresentamos agora um programa bastante útil com algumas características interessantes.

Se ainda não tiver encontrado uma função definida pelo utilizador, a linha 20 talvez o preocupe um pouco. É lida do seguinte modo: "Defina a função A de X como..." Por outras palavras, sempre que quisermos podemos atribuir a uma variável o valor obtido por uma função por nós introduzida, não nos limitando apenas às previstas na versão de Basic usada pela máquina. Na prática, isto permite poupar espaço de memória. A função que aqui usamos atribui um valor aleatório na gama A a M + A, sendo chamada duas vezes na linha 1000. Se a sua máquina não dispuser desta instrução (DEF FN), omita muito simplesmente a linha 20 e escreva:

```
1000 N1 = INT(RND(0)*M + A):N2 = INT(RND(0)*M + A)
```

A segunda coisa a notar quanto a este programa é o "menú" das linhas 40 a 80. Trata-se de um estratagema vulgar e muito útil quando se trata de apresentar ao utilizador uma escolha entre várias hipóteses. Neste caso numerei as hipóteses, devendo o utilizador escrever um número em resposta; mas se se quiser é possível obrigar o utilizador a escrever um nome, ou talvez apenas uma inicial.

O uso das variáveis M e A permite-nos manipular a gama de números apresentados, e será fácil compreender que estes dependem do nível de dificuldade escolhido. M é um multiplicador do nível de dificuldade, enquanto que A é um factor a somar. Este último é particularmente importante para a divisão, pois sem ele seria possível dividir por zero, levando o programa a ser interrompido.

Finalmente, e pela primeira vez neste livro, iremos usar subrotinas. Estas permitem poupar muito espaço de memória. A escolha de números e a sua apresentação são rotinas comuns a todas as operações escolhidas, e portanto é lógico dispor de uma subrotina apesar de ser verdade que, pelo menos a este nível, teria sido possível usar GOTOs para eliminar ambas as subrotinas. No entanto, constitui um bom hábito de programação usar subrotinas sempre que possível. A listagem torna-se mais fácil de ler e a eliminação de erros de programação é muito facilitada.

Certos autores afirmam que uma rotina deve ter apenas uma saída, mas pessoalmente não tenho a certeza disto (veja as linhas 2070 e 2090). Estes autores prefeririam que em vez de incluir RETURN no final destas linhas tivesse escrito GOTO 2100. Esta linha conteria assim a simples instrução RETURN, sendo de facto a única saída da subrotina. Pessoalmente, penso que os GOTOs são horríveis e tento evitá-los sempre que possível.

Lista de variáveis

T	Tipo de problema escolhido
D	Nível de dificuldade seleccionado
M	Multiplicador aleatório
A	Factor aleatório a adicionar
S	Problema em estudo
R	Resultado correcto dos cálculos
C	Número de soluções correctas indicadas
I	Entradas do utilizador
N1	Primeiro número aleatório
N2	Segundo número aleatório
Q$	String de resposta

Listagem do programa

```
0010 PRINT :PRINT "ARITMÉTICA"
0020 DEF FNA(X) = INT(RND(0)*M + A)
0030 PRINT :PRINT "Olá! Estou programado para apresen
tar-lhe 20 resultados."
0040 PRINT :PRINT "De que tipo prefere —"
```

```
0050 PRINT "1 — somar"
0060 PRINT "2 — diminuir"
0070 PRINT "3 — multiplicar"
0080 PRINT "4 — dividir...."
0090 INPUT T:IF T>4 THEN 110
0100 IF T>0 THEN 120
0110 PRINT "Por favor escreva um número entre 1 e 4":
GOTO 90
0120 PRINT "Dificuldade: escreva um número entre 1 (fácil)
e 9 (difícil)"
0130 INPUT D:IF D>9 THEN 150
0140 IF D>0 THEN 160
0150 PRINT "Um número entre 1 e 9, por favor.":GOTO
130
0160 PRINT
0170 PRINT "Certo, cá vamos... ":C = 0
0180 FOR S = 1 TO 10:ON T GOTO 200,300,400,500
0190 REM — adição + + + + + + + + + + + + + + + + +
0200 M = 10*D:A = 10*(D-1):GOSUB 1000
0210 R = N1 + N2:GOSUB 2000
0220 GOTO 600
0290 REM — subtracção + + + + + + + + + + + + + +
0300 M = 10*D:A = 10*(D-1):GOSUB 1000
0310 R = N1:N1 = R + N2:GOSUB 2000
0320 GOTO 600
0390 REM — multiplicação + + + + + + + + + + + +
0400 M = D + 3:A = 0:GOSUB 1000
0410 R = N1*N2:GOSUB 2000
0420 GOTO 600
0490 REM — divisão + + + + + + + + + + + + + + +
0500 M = D + 3:A = 1:GOSUB 1000
0510 R = N1:N1 = R*N2:GOSUB 2000
0600 NEXT S
0610 IF C<20 THEN 630
0620 PRINT: PRINT :PRINT "PARABÉNS! — ";
0630 PRINT "A sua pontuação foi ";C;" num total de 20."
0640 INPUT "Quer jogar de novo",Q$
0650 IF LEFT$(Q$,1) = "S" THEN 40
0660 PRINT :PRINT
0670 PRINT "Adeus — foi divertido jogar consigo."
0680 FOR X = 1 TO 10:PRINT:NEXT X:END
0990 REM — Escolher dois números + + + + + + + + +
1000 N1 = FNA(X):N2 = FNA(X):RETURN
1990 REM — Imprimir problema, aceitar e verificar respos-
ta + + + + + + + + + + + + + + +
2000 PRINT :PRINT
2010 PRINT "Quanto é ";N1;
2020 IF T = 1 PRINT " + ";
2030 IF T = 2 PRINT " - ";
2040 IF T = 3 PRINT "X";
2050 IF T = 4 PRINT "dividido por";
2060 PRINT N2;:INPUT I
2070 IF I = R PRINT "/CERTO!":C = C + 1:RETURN
2080 PRINT " É pena, está errado."
2090 PRINT "A resposta correcta é ";R:RETURN
```

Exemplo de execução

```
Olá! Estou programado para apresentar-lhe 20 resultados.
De que tipo prefere —
1 — somar
2 — diminuir
3 — multiplicar
4 — dividir...?1
Dificuldade: escreva um número entre 1 (fácil) e 9 (difícil)
3
Certo, cá vamos...

Quanto é 30 + 44? 74 / CERTO!

Quanto é 42 + 25? 67 / CERTO!

Quanto é 36 + 34? 70 / CERTO!

Quanto é 46 + 26? 68   É pena, está errado.

A resposta correcta é 72
```

Quanto é 40 + 32? 72 / CERTO!

Quanto é 27 + 21? 48 / CERTO!

Quanto é 37 + 40? 77 / CERTO!

Quanto é 38 + 44? 72 É pena, está errado.
A resposta correcta é 82

Quanto é 42 + 25? 67 / CERTO!

Quer jogar de novo? N
Adeus — foi divertido jogar consigo

XII

SÉRIES DE NÚMEROS (3 K)

As séries de números são um bom teste de atenção. São apresentados ao utilizador alguns números por uma dada ordem, sendo-lhe pedido que indique o ou os números seguintes em função da lógica dessa ordenação. Por exemplo, considere a série seguinte:

2 4 6 8 10 12

É óbvio que o número seguinte é 14. Que fizemos? Notámos que a diferença entre o primeiro e o segundo número é de 2, entre o segundo e o terceiro também 2, e deduzimos que daí em diante todas as diferenças serão iguais. Verificamos esta hipótese observando as diferenças restantes. Como são de facto iguais a 2, concluímos que se trata de uma regra invariável e podemos extrapolar para o par de números que se segue.

Se a série de números apresentada pelo computador fosse sempre do mesmo tipo, o teste tornar-se-ia extremamente maçador. Se, por exemplo, deduzirmos que as diferenças entre números são sempre as mesmas para uma dada série, basta-nos olhar para os dois primeiros elementos para descobrir qual a diferença, adicionando-a ao último. Não se pode dizer que seja muito divertido... Mas consideremos o seguinte exemplo:

2 4 7 11 16 22

Já temos de fazer uma pausa até compreendermos que o incremento de um número para o seguinte também é aumentado (a propósito, o número seguinte é 29...)

Do que se disse deve poder concluir-se que para ter êxito o programa deve ser capaz de apresentar muitos tipos de séries diferentes. O programa que apresentamos a seguir faz exactamente isso — 10 tipos diferentes — e não será difícil aumentar ainda mais as hipóteses se o leitor desejar.

O tipo de problema é escolhido na linha 110, sendo executada uma nova selecção na linha 7010 em alguns casos. Utilizo três funções definidas por mim designadas por A, B e C, que fornecem números aleatórios até 9, 99 e 7 respectivamente. Se não puder usar DEF FNs, empregue em vez delas as instruções de selecção aleatória que a sua máquina aceita sempre que vir FN na listagem.

Na linha 90 defini um comprimento de linha de 80 caracteres porque a tripla sequência produz linhas compridas que podem passar para a linha seguinte, talvez cortando um número de dois ou três algarismos ao meio; o que obviamente não é aconselhável. Do mesmo modo, na linha 150 voltei ao comprimento habitual para o uso em programas subsequentes.

Se o leitor conceber uma série nova, inclua uma pequena subrotina que a trate e em seguida altere a linha 110.

Lista de variáveis

Q$	String de resposta
P	Número da série apresentada
T	Tipo de série
S	Pontuação
X	Contador dos elementos da série
Z	Solução correcta
Q	Primeiro algarismo indicado pelo utilizador
Q1	Segundo algarismo indicado pelo utilizador
A a F	Números para uso na série.

Listagem do programa

```
0010 PRINT TAB(20);"SÉRIES"
0020 PRINT TAB(20);"= = = ="
0030 PRINT :INPUT "Deseja instruções (S/N)",Q$
0040 IF LEFT$(Q$,1) = "S" THEN GOSUB 9000
0050 DEF FNA(X) = INT(RND(0)*10)
0060 DEF FNB(X) = INT(RND(0)*100)
0070 DEF FNC(X) = INT(RND(0)*7) + 1
0080 PRINT :PRINT
0090 LINE = 80
0100 FOR P = 1 TO 20
0110 T = FNC(T).ON T GOSUB 1000, 2000, 3000, 4000,
5000, 6000, 7000
0120 PRINT :PRINT:NEXT P
0130 PRINT :PRINT:PRINT
0140 PRINT "A SUA PONTUAÇÃO FOI ";S;"NUM TO
TAL DE 20."
0150 LINE = 64:END
1000 REM — Sequência simples
1010 A = FNA(A):IF RND>.5 THEN A = -A
1020 B = FNB(B):IF RND>.5 THEN B = -B
1030 FOR X = 1 TO 6
1040 PRINT B + X*A;", ";:NEXT X:PRINT
1050 Z = B + 7*A
1060 INPUT "QUE VEM A SEGUIR",Q
1070 IF Q = Z PRINT "CORRECTO":S = S + 1:RETURN
1080 PRINT "É PENA — DEVIA SER ";Z:RETURN
2000 REM — Incremento progressivo
2010 A = FNA(A):IF RND>.5 THEN A = -A
2020 B = FNB(B):IF RND>.5 THEN B = -B
2030 C = 1: IF RND>.5 THEN C = -1
2040 FOR X = 1 TO 6
2050 PRINT B;", ";:B = B + A:A = A + C:NEXT X:PRINT
2060 Z = B:GOTO 1060
3000 REM — Alternadas
3010 A = FNA(A):B = FNA(A):IF A = B THEN 3010
3020 C = FNA(A):D = FNA(A)
3030 FOR X = 1 TO 4
3040 PRINT A + X*C;", ";B + X*D;", ";:NEXT X:PRINT
3050 GOSUB 8000:A = A + 5*C:B = B + 5*D:GOTO 5090
4010 A = FNA(A):B = FNA(B)
4020 C = FNA(A):D = -(FNA(D))
4030 GOTO 3030
5010 A = FNA(A):B = FNA(A):IF A = B THEN 5010
5020 C = FNAB(C):IF C = A THEN 5020
5030 IF C = B THEN 5020
5040 I = FNC(I)
5050 FOR X = 1 TO 4
```

```
5060 PRINT  A;",  ";B;",  ";C;",  ";:A=A+I:B=B+I:-
C=C+I:NEXT C: PRINT
5070 GOSUB 8000
5090 IF Q<>A THEN 5110
5100 IF Q1=B THEN 5120
5110 PRINT "NÃO. DEVIA SER ";A;" E ";B:RETURN
5120 PRINT "CORRECTO":S=S+1:RETURN
6010 A=FNA(A):IF RND>.5 THEN A=-A
6020 B=FNA(A):IF RND>.5 THEN B=-B
6030 IF A=B THEN 6020
6040 C=INT(RND(0)*3):IF RND>.5 THEN C=-C
6050 D=INT(RND(0)*2+1):IF RND>.5 THEN D=-D
6060 E=1:IF RND>.5 THEN E=-1
6062 F=1:IF RND>.5 THEN F=-1
6065 FOR X=1 TO 5
6070 PRINT   A;",   ";B;",   ";:A=A+C:B=B+D:-
C=C+E:D=D+F
6080 NEXT X:PRINT
6090 GOSUB 8000:GOTO 5090
7000 REM — Casos especiais
7010 ON INT(RND(0)*4+1)GOTO 7100,7200,7300,7400
7100 A=INT (RND(0)*4+1):FOR X=1 TO 6
7110 PRINT A;", ";:A=A*2:NEXT X:PRINT
7120 Z=A:GOTO 1060
7200 R=FNC(A):FOR X=1 TO 6
7210 PRINT A*A;", ";:A=A+1:NEXT X:PRINT
7220 Z=A*A:GOTO 1060
7310 A=FNA(A):IF RND>.5 THEN A=-A
7320 B=FNA(A):IF A=B THEN 7320
7325 IF B<A THEN C=A:A=B:B=C
7330 FOR X=1 TO 7
7340 PRINT   A;",   ";:C=A:A=B:B=A+C:NEXT   X:-
PRINT
7350 Z=A:GOTO 1060
7410 A=0:FOR  D=1  TO  4:B=C:C=FNA(C)+1:A=-
A*10+C:NEXT D
7420 B=C:C=1:IF RND>.5 THEN C=-1
7430 FOR X=1 TO 7:PRINT A;", ";
7440 B=B+C:A=(A-INT(A//1000)*1000)*10+B
7450 NEXT X:PRINT:Z=A:GOTO 1060
```



```
8000 INPUT "QUE VEM A SEGUIR",Q
8010 INPUT "E DEPOIS",Q1
8020 RETURN
9000 PRINT "Todos conhecemos muitas séries de números, "
9010 PRINT "como por exemplo 1 2 3 4 5 ou 3 6 9 12 15 ";
9020 PRINT "e sabemos indicar o número que se segue na ";
9030 PRINT "série. Este programa é desse tipo, se bem ";
9040 PRINT "que evidentemente nem todas as séries sejam ";
9050 PRINT "tão simples! indico os primeiros números, e ";
9060 PRINT "você deve indicar o seguinte."
9070 PRINT :INPUT "Carregue em RETURN para come
çar...",Q$:RETURN
```

Exemplo de execução

```
Deseja instruções (S/N)? N
-63 ,-61 , -60 , -60 , -61 , -63,
QUE VEM A SEGUIR? -60
É PENA — DEVIA SER -66

11 , 28 , 18 , 21 , 25 , 14 , 32 , 7 ,
QUE VEM A SEGUIR? 39
E DEPOIS? 0
CORRECTO

7 , 3 , 82 , 13 , 9 , 88 , 19 , 15 , 94 , 25 , 21 , 100 ,
QUE VEM A SEGUIR? 31
E DEPOIS?27
CORRECTO

21 , 16 , 12 , 9 , 7 , 6 ,
QUE VEM A SEGUIR? 6
CORRECTO

-60 , -68 , -75 , -81 , -86 , -90 ,
QUE VEM A SEGUIR? -93
CORRECTO
```

10 , 10 , 14 , 19 , 18 , 28 , 22 , 37 ,
QUE VEM A SEGUIR? 26
E DEPOIS? 46
CORRECTO

25 , 36 , 49 , 64 , 81 , 100 ,
QUE VEM A SEGUIR? 121
CORRECTO

7 , 1 , 41 , 9 , 3 , 43 , 11 , 5 , 45 , 13 , 7 , 47 ,
QUE VEM A SEGUIR? 15
E DEPOIS? 9
CORRECTO

XIII

CORRIDAS DE CAVALOS (3 K)

Já foram escritos muitos programas de corridas de cavalos, mas o que torna este um pouco diferente é o sistema de apostas usado, que se baseia na diferente probabilidade de vencer de cada um dos cavalos. Esta probabilidade varia em função do comportamento do cavalo, começando inicialmente por 4 para 1 em todos os casos. Depois, à medida que é ganha uma corrida, o vencedor vê as suas probabilidades diminuirem enquanto as dos outros aumentam.

Um refinamento deste programa consistiria em dispor de um conjunto de cavalos (por exemplo 10 ou 12) de entre os quais só 6 seriam escolhidos para cada corrida. Outra possibilidade ainda consistiria em atribuir parâmetros a cada cavalo em função das suas potencialidades, tendo estes parâmetros em conta em cada movimento. Estes parâmetros devem obviamente ser bastante pequenos — certamente nunca superiores a 3 ou 4 — e devem ser reduzidos à medida que o cavalo ganha corridas. Isto equivaleria portanto ao "cansaço" do animal. Estes dois refinamentos colocariam curiosos dilemas aos apostadores que tivessem de escolher entre dois cavalos que ainda não tivessem corrido juntos.

Mesmo sem estes refinamentos, o jogo é bastante divertido e pode provocar bastante excitação. Os jogadores podem apostar tanto quanto quiserem em qualquer cavalo, e o jogo termina quando um dos jogadores fica sem dinheiro.

O visor apresenta a corrida em diferentes fases, mas se o seu terminal pode ser comutado para modo página ou modo "scroll" a corrida pode ser executada em modo página com o cursor desligado. O resultado é uma corrida bastante animada, muito interessante de observar.

O programa utiliza duas matrizes sob a forma de tabelas, uma para vários parâmetros relacionados com os cavalos e a outra para os jogadores. Os pormenores são indicados na lista de variáveis. Note que no interior dos parêntesis o primeiro algarismo indica um cavalo ou jogador, enquanto que o segundo indica um dado ligado a esse cavalo ou jogador. Este é talvez o método mais comum de uso de uma matriz, e simplifica bastante a programação. Note no entanto que a maior parte das BASICs reservam memória para uma matriz de dimensões 10 por 10 a menos que se especifiquem outras dimensões, pelo que o dimensionamento das matrizes na linha 100 permite poupar uma enorme quantidade de memória.

Encontramos igualmente um array P$ onde se guarda o nome de cada jogador de tal modo que o computador se possa dirigir a este usando o seu nome, o que torna o programa mais agradável. Este array foi deixado não dimensionado, pelo que podem jogar simultâneamente 10 pessoas. Se lhe parece que este número é excessivo, diminua-o acrescentando uma instrução DIM no final da linha 40.

Lista de variáveis

```
P        Número total de jogadores
P$(P)    Nome do jogador P
H(X,Y)   Parâmetros dos cavalos, onde
            X = número do cavalo
            Y = 1 = posição na tabela ou corrida
                2 = probabilidade de vencer
P(X,Y)   Parâmetros dos jogadores, onde
            Y = 1 = Dinheiro disponível
                  2 = Dinheiro apostado
                  3 = Cavalo em que se aposta
B        Actual aposta do jogador
H        Actual número do cavalo
X        Contador
F        Posição na tabela do cavalo mais rápido
Q        Número do vencedor; também contador
W        Quantidade de vitórias
Q$       String de entradas
```


Listagem do programa

```
0010 PRINT :PRINT:PRINT TAB (10); "CORRIDAS DE
CAVALOS"
0020 PRINT
     TAB(10);"= = = = = = = = = = = = = = = = = = ="
0030 PRINT :PRINT "POR FAVOR INDIQUE NOMES"
0040 P=1
0050 PRINT "Jogador n.";:INPUT P$ (P)
0060  IF P$(P)=""THEN 80
0070 P=P+1:GOTO 50
0080 P=P−1:PRINT
0090 PRINT "Cada jogador começa com $100.":PRINT
0100 DIM H(6,2),P(P,3)
0110 FOR X=1 TO P:P(X,1)=100:NEXT X
0120 PRINT "Todos os cavalos começam a 4 para 1"
0130 FOR X=1 TO 6:H(X,2)=4
0140 NEXT X
0150 GOSUB 2000
0160 FOR X=1 TO P
0170 P$(X);" — você tem $";P(X,1);". Quanto quer apostar";
0180 INPUT B:IF B<=P(X,1) THEN 200
0190 PRINT "Não tem esse dinheiro!":GOTO 170
0200 P(X,2)=B:P(X,1)=P(X,1)−B
0210 INPUT "Em que cavalo",B
0220 IF B<=6 THEN 240
0230 PRINT"Esse cavalo não participa na corrida":GOTO 210
0240 IF B<=0 THEN 230
0250 P(X,3)=B:NEXT X
0260 REM Modo página. CLS e imprimir meta
0270 PRINT CHR$(25);CHR$(25);CHR$(25);CHR$(5)
0280 FOR X=1 TO 6:PRINT TAB(40):"!":NEXT X
0290 FOR X=1 TO 6:H(X,1)=1:NEXT X
0300 PRINT "PARTIRAM!"
0310 PRINT CHR$(16):REM reset cursor
0320 FOR X=1 TO 6
0330 FOR Z=1 TO H(X,1)
0340 PRINT ".";:NEXT Z
0350 PRINT X: NEXT X
0360 REM Verificar vencedor
```

```
0370 Q=0:F=0:FOR X=1 TO 6
0380 IF H(X,1)<39 THEN 400
0390  IF H(X,1)>F THEN F=H(X,1):Q=X
0400 NEXT X
0410 IF Q<>0 THEN 500
0420 REM Mover cavalos
0430 FOR X=1 TO 6
0440 H(X,1)=H(X,1)+INT(RND(0)*7)+1
0460 NEXT X:GOTO 310
0500 PRINT :PRINT "O VENCEDOR É O NÚMERO
";Q;"!"
0510 PRINT "VIVA   ";:GOSUB 3000:PRINT "!"
0520 PRINT :PRINT
0530 REM — Modo scrolling
0540 PRINT CHR$(14)
0550 REM — Relatório aos jogadores
0560 FOR X=1 TO P
0570 PRINT P$(X);" — a sua aposta no n.";P(X,3);
0580 IF Q<>P(X,3) THEN PRINT "PERDEU":GOTO 630
0590 PRINT "ganhou ";H(Q,2);"para 1"
0600 W=P(X,2)*H(Q,2)
0610 PRINT "GANHOU $";W;" ALÉM DO QUE APOS
TOU."
0620 P(X,1)=P(X,1)+W+P(X,2):GOTO 640
0630 P(X,2)=0:P(X,3)=0
0640 NEXT X:PRINT:PRINT
0650 PRINT "POSIÇÃO FINANCEIRA GLOBAL — "
0660 FOR X=1 TO P
0670 PRINT  P$(X);" $";P(X,1)
0680 NEXT X
0685 FOR X=1 TO P:IF P(X,1)=0 THEN 4000
0686 NEXT X
0690 PRINT
0700 INPUT "Outra corrida",Q$
0710  IF LEFT$ (Q$,1)<>"Y" THEN END
0720 REM — Ajustar probabilidades
0730 FOR X=1 TO 6
0740 IF X=Q THEN H(X,2)=INT(H(X,2)/2)+1:GOTO 760
0750 H(X,2)=H(X,2)+1
0760 NEXT X
```



```
0770 GOSUB 2000:GOTO 160
2000 REM Indicar probabilidades
2010 FOR Q=1 TO 6
2020 PRINT Q;" — ";:GOSUB 3000
2030 PRINT " ";H(Q,2);"PARA 1"
2040 NEXT Q
2050 PRINT
2060 RETURN
3000 ON Q GOTO 3010,3020,3030,3040,3050,3060
3010 PRINT "LIVELY LADY":GOTO 3070
3020 PRINT "OWZYAFARVER";:GOTO 3070
3030 PRINT "SPINDRIFT";:GOTO 3070
3040 PRINT "RUNNING WILD";:GOTO 3070
3050 PRINT "UNCLE VANYA";:GOTO 3070
3060 PRINT "WHITE TORNADO";:GOTO 3070
3070 RETURN
4000 PRINT "JOGADOR(ES) SEM DINHEIRO. FIM DO
JOGO"
```

Exemplo de execução

```
POR FAVOR INDIQUE NOMES
Jogador n.1 ? João
Jogador n.2 ? Maria
Jogador n.3 ? Manuel
Jogador n.4 ?
Cada jogador começa com $100

Todos os cavalos começam a 4 para 1
1 — LIVELY LADY 4 PARA 1
2 — OWZYAFARVER 4 PARA 1
3 — SPINDRIFT 4 PARA 1
4 — RUNNING WILD 4 PARA 1
5 — UNCLE VANYA 4 PARA 1
6 — WHITE TORNADO 4 PARA 1

João — você tem $100. Quanto quer apostar? 10
Em que cavalo? 5
Maria — você tem $100. Quanto quer apostar? 50
Em que cavalo? 3
```

Manuel — você tem $100. Quanto quer apostar? 20
Em que cavalo? 4
PARTIRAM!

O VENCEDOR É O NÚMERO 5!
VIVA UNCLE VANYA!

João — a sua aposta no n.5 ganhou 4 para 1
GANHOU $40 ALÉM DO QUE APOSTOU.
Maria — a sua aposta no n.3 perdeu
Manuel — a sua aposta no n.4 perdeu

POSIÇÃO FINANCEIRA GLOBAL —
João $140
Maria $50
Manuel $80

Outra corrida? SIM
1 — LIVELY LADY 5 PARA 1
2 — OWZYAFARVER 5 PARA 1
3 — SPINDRIFT 5 PARA 1
4 — RUNNING WILD 5 PARA 1
5 — UNCLE VANYA 3 PARA 1
6 — WHITE TORNADO 5 PARA 1

XIV

NUCLEAR (3,5 K)

"Nuclear" é um jogo de perícia entre dois jogadores. É suficientemente explicado nas instruções.

Este programa também utiliza uma matriz, mas desta vez como área de jogo. O leitor notará que eu dimensionei uma matriz de 8×8, atribuindo o valor zero a todos os seus elementos, mas que apenas usei uma 6×6. Fi-lo por uma questão de simplicidade na escrita do programa, porque obtenho assim um rebordo sem nada à volta da rede onde se efectua o jogo. O interesse disto torna-se óbvio quando se observam as linhas 1140 a 1160: foi encontrado um átomo em M(I,J), e é necessário verificar se possui ou não três ou mais vizinhos. Se o ponto I,J estivesse na periferia da matriz, X e/ou Y podia estar fora dela; o computador teria de verificar um elemento não existente na matriz e o programa falharia. Como é óbvio, o problema poderia ser resolvido usando instruções IF do tipo "IF X<1 THEN...", mas como seriam ocupadas quatro linhas, além do facto de nas linhas 1260 a 1270 surgir uma rotina semelhante, decidi-me a não as usar. Este método é aliás particularmente simples.

À medida que cada jogador coloca um átomo na matriz, esse elemento da matriz é passado para um valor correspondente ao jogador — 1 para A e 2 para B. O elemento pode passar igualmente para o valor 10 quando explode. Esta secção ocupa as linhas 1200 a 1310, onde é verificada a flag C. A flag inicia o ciclo com o valor 0, passando a 1 quando é feita qualquer alteração; isto é, quando explode um átomo. Nestas condições, se C tem ainda o valor 0 quando é atingida a linha 1310, a reacção em cadeia termina.

É indicado aos jogadores quais os átomos que explodiram e a linha 1380 limpa os restos — isto é, passa a zero o elemento da matriz. O resto da listagem é fácil de compreender.

O jogo pode prosseguir durante bastante tempo entre dois jogadores bons. A estratégia a utilizar não é fácil de descobrir, se bem que os cantos sejam mais seguros do que os lados e estes melhores do que o meio da rede. Como é óbvio, torna-se necessário espalhar os átomos pela rede a fim de não os colocar em cadeia.

Lista de variáveis

Q$	String de entradas
M(X,Y)	Rede de jogo, com margem em redor
X	Contador de linha
Y	Contador de coluna
I	Contador principal de linha
J	Contador principal de coluna
P	Jogador actual
F	Flag de final
N	Contagem de átomos vizinhos
C	Flag de alteração ou explosão
A	Contador de átomos do jogador A
B	Contador de átomos do jogador B

Listagem do programa

```
0010 PRINT "NUCLEAR"
0020 PRINT "———————"
0030 PRINT
0040 INPUT "Deseja instruções",Q$
0050 IF LEFT$ (Q$,1) = "Y" THEN 9000
0055 REM limpar matriz
0060 DIM M(8,8)
0070 FOR X = 1 TO 8:FOR Y = 1 TO 8
0080 M(X,Y) = 0
0090 NEXT Y: NEXT X
0095 REM loop de jogo continua indefinidamente
0100 FOR P = 1 TO 2:GOSUB 1000
0110 IF F<>0 THEN 0130
0120 NEXT P:GOTO 100
0130 PRINT :PRINT
```



```
0140 ON F GOTO 150,160
0150 PRINT "JOGO EMPATADO":GOTO 180
0160 IF P = 1 PRINT "A";
0170 IF P = 2 PRINT"B";
0180 PRINT :PRINT:END
0990 REM — Vez do jogador
1000 GOSUB 2000:REM imprimir mapa
1005 PRINT "Jogador ";:IF P = 1 PRINT "A";
1010 IF P = 2 PRINT "B";
1015 INPUT " — O SEU ÁTOMO (LINHA,COLUNA) ",I,J
1020 IF I>6 THEN 1060
1030 IF I<1 THEN 1060
1040 IF J<1 THEN 1060
1050 IF J< = 6 THEN I = I + 1:J = J + 1:GOTO 1070
1060 PRINT "Movimento ilegal":GOTO 1070
1070 IF M(I,J)<>0 PRINT "Essa posição já está ocupa
da":GOTO 1010
1080 M(I,J) = P
1090 REM Verificar estado crítico
1100 REM Para cada átomo, contar vizinhos
1110 FOR I = 2 TO 7:FOR J = 2 TO 7
1120 IF M(I,J) = 0 THEN 1180
1130 N = 0:REM Contagem de vizinhos
1140 FOR X = I−1 TO I + 1:xFOR Y = J−1 TO J + 1
1150 IF M(X,Y)<>0 THEN N = N + 1
1160 NEXT Y:NEXT X
1170 IF N< = 4 THEN 1200
1180 NEXT J:NEXT I
1190 RETURN :REM — Nenhum crítico
1195 REM Início da reacção em cadeia
1200 PRINT :PRINT "Massa em ";I−1;J−1; "passa a crítica"
1210 M(I,J) = 10
1220 REM Para cada crítico, afectar vizinhos
1230 C = 0:FOR I = 2 TO 7:FOR J = 2 TO 7
1240 IF M(I,J)<>10 THEN 1300:REM Descobrir um crítico
1250 FOR X = I−1 TO I + 1: FOR Y = J = 1 TO J + 1
1260 IF M(X,Y) = 1 THEN 1280
1270 IF M(X,Y)<>2 THEN 1290
1280 M(X,Y) = 10:C = 1:REM Alterar flag
1290 NEXT Y:NEXT X
```

```
1300 NEXT J:NEXT I
1310 IF C=0 THEN 1330
1320 PRINT "A reacção continua":GOTO 1230
1330 PRINT :PRINT
1340 PRINT "A reacção terminou":GOSUB 2000
1350. REM Limpar restos, verificar resultados
1360 A=0:B=0
1370 FOR I=2 TO 7:FOR J=2 TO 7
1380 IF M(I,J)=10 THEN M(I,J)=0
1390 IF M(I,J)=1 THEN A=A+1
1400  IF M(I,J)=2 THEN B=B+1
1410 NEXT J:NEXT I
1420 IF A<>0 THEN 1450
1430 IF B=0 THEN F=1:RETURN:REM Jogo empatado
1440 P=2:F=2: RETURN:REM B ganha
1450 IF B<>0 THEN F=0:RETURN REM — Jogo continua
1460 P=1:F=2:RETURN:REM A ganha
1990 REM Imprimir mapa
2000 PRINT
2010 PRINT "      1 2 3 4 5 6"
2020 FOR I=2 TO 7:PRINT I-1;" ";
2030 FOR J=2 TO 7
2040 IF M(I,J)=0 PRINT ".";
2050 IF M(I,J)=1 PRINT "A";
2060 IF M(I,J)=2 PRINT "B";
2065 IF M(I,J)=10 PRINT "*";
2070 PRINT " ";:NEXT J
2080 PRINT:NEXT I:PRINT:RETURN
9000 PRINT "'NUCLEAR' É UM JOGO ENTRE DOIS JO
GADORES, ";
9010 PRINT "CADA UM COLOCANDO UM ÁTOMO DE
URÂNIO 235 NUMA ";
9020 PRINT "REDE. QUALQUER ÁTOMO COM 3 OU
MAIS VIZINHOS ";
9030 PRINT "TORNA-SE INSTÁVEL E EXPLODE,
DESTRUINDO-SE A ";
9040 PRINT "SI MESMO E AOS VIZINHOS, QUE POR
SUA VEZ DESTROEM ";
9050 PRINT "OUTROS. ISTO PROVOCA UMA REAC
ÇÃO EM CADEIA. O ";
```



```
9060 PRINT "VENCEDOR É AQUELE QUE DESTRÓI
TODOS OS ÁTOMOS ";
9070 PRINT "DO OPOSITOR MANTENDO UM OU
MAIS DOS SEUS."
9080 INPUT "CARREGUE EM RETURN PARA COME
ÇAR...",Q$
9090 RETURN
```

Exemplo de execução

Deseja instruções? NÃO.

Jogador A — O SEU ÁTOMO (LINHA,COLUNA)? 1 ,4

Jogador B — O SEU ÁTOMO (LINA,COLUNA)? 3 ,6

Jogador A — O SEU ÁTOMO (LINHA,COLUNA)? 6 ,6

Jogador B — O SEU ÁTOMO (LINHA,COLUNA)? 3 ,3

Jogador A — O SEU ÁTOMO (LINHA,COLUNA)? 4 ,4

Jogador B — O SEU ÁTOMO (LINHA,COLUNA)? 4 ,1

Jogador A — O SEU ÁTOMO (LINHA,COLUNA)? 5 ,4

Jogador B — O SEU ÁTOMO (LINHA,COLUNA)? 3 ,3

Massa em 3 3 passa a crítica

A reacção continua

A reacção continua

A reacção terminou

Jogador A — O SEU ÁTOMO (LINHA,COLUNA)? 3 ,5

Jogador B — O SEU ÁTOMO (LINHA,COLUNA)? 5 ,5

Jogador A — O SEU ÁTOMO (LINHA,COLUNA)? 3 ,3

Jogador B — O SEU ÁTOMO (LINHA,COLUNA)? 4 ,4

Massa em 4 4 passa a crítica.

A reacção continua

A reacção terminou

XV

ESCREVER NÚMEROS (2 K)

Uma subrotina pode chamar outra. Isto coloca imediatamente duas questões: como se sai da subrotina, e qual será o interesse dela?

A primeira questão não é difícil. Basta contar o número de chamamentos de rotinas e assegurar a existência de um número equivalente de instruções RETURN. O objectivo de uma rotina deste tipo, por outro lado, consiste em poupar espaço de memória quando é necessário repetir as mesmas operações ou a mesma sequência de operações repetidamente. Este programa faz exactamente isto.

Quando um ser humano observa um algarismo como 12345, e o traduz por "doze mil trezentos e quarenta e cinco", passou os olhos sobre o número da direita para a esquerda, dividindo-o em grupos de três. Cada grupo de três tem um nome — milhões, milhares — e dentro de cada grupo os algarismos são novamente lidos da direita para a esquerda para determinar as centenas, as dezenas e as unidades, assim como o mais elevado par de algarismos que pode ser expresso por uma única palavra — doze, dezoito. Este processo é repetido para cada grupo de três, acrescentando em seguida o nome aplicado a cada grupo.

Para fazer o mesmo trabalho, um programa de computador deve seguir as mesmas regras; e como é necessário aplicar sempre o mesmo método para cada grupo, o programa é feito com base em subrotinas.

Este programa imprimirá em palavras o valor de qualquer número. Pode tratar até 14 caracteres, mas faz um arredondamento no nono algarismo. As fracções decimais são tratadas até nove casas decimais, sujeitas ao mesmo arredondamento.

Este programa foi desenvolvido quando um pai exprimiu a sua preocupação devido à dificuldade do filho ler números; mas pode ter outras utilidades, em particular no campo educativo. A sua parte mais importante é, evidentemente, a subrotina entre as linhas 9000 e 9490, que pode ser retirada completa e inserida em qualquer programa que exija este tipo de apresentação dos números. Mesmo que o leitor nunca venha a usar este programa, vale a pena estudá-lo, quanto mais não seja para "sentir" como funcionam estas subrotinas.

A linha 120 determina se está ou não presente uma fracção, saltando para a linha 200 se tal não acontecer. Se existir uma fracção, a linha 130 coloca a parte inteira na variável N e transforma o resto na string F$. É tratada a variável N, e em seguida, nas linhas 150 a 180, a parte decimal é lida, um algarismo de cada vez.

Vejamos agora a subrotina principal. N contém o número a imprimir, e as linhas 9010 a 9030 tratam-na em função das suas dimensões (se observar o último exemplo de execução, notará que também é usada notação científica). Se estudar cuidadosamente esta subrotina verificará que, daí em diante, o programa elimina factores de dez a fim de descobrir o maior número que pode ser expresso por uma palavra única. Consideremos o exemplo 12345. A linha 9030 provocará um salto para a linha 9180, onde N é armazenado em S, a memória dos "milhares". N recebe o valor 12 e a subrotina chama-se a si mesma.

Desta vez, a linha 9050 provoca um salto para 9300, a partir da qual se realiza a impressão da palavra "doze". A RETURN da linha 9340 provoca um retorno à instrução que chamou a rotina — neste caso a linha 9200, que imprime a palavra "mil". O valor original de N é então recuperado da variável S; diminuem-se os milhares, e repete-se todo o processo.

Recomendo novamente ao leitor que execute este programa várias vezes, usando números diferentes. A experiência dir-lhe-á mais sobre o uso de subrotinas do que um milhão de palavras.

Lista de variáveis

N	Número a imprimir
N1	Memória temporária de N
F$	String da fracção decimal
F	Comprimento de F$
X	Contador digital
B	Memória dos biliões
M	Memória dos milhões
S	Memória dos milhares
H	Memória das centenas
D	Contador geral
A$	Palavra a imprimir

Listagem do programa

```
0010 PRINT TAB(15),"ESCREVER NÚMEROS"
0020 PRINT TAB(15),"= = = = = = = = = = = ="
0030 PRINT
0040 PRINT "Este programa imprimirá em palavras o valor ";
0050 PRINT "de qualquer número. São tratados até 14 ";
0060 PRINT "algarismos, com um arredondamento no nono. ";
0070 PRINT "As fracções decimais são tratadas até à ";
0080 PRINT "nona casa, com o mesmo arredondamento."
0100 PRINT :INPUT "Escreva um número",N
0120 IF N=INT(N) 200
0130 N1=INT(N):F$=STR$(N-N1):F=LEN(F$):N=N1
0140 GOSUB 9000:PRINT " vírgula ";
0150 FOR X=3 TO F
0160 IF MID$(F$,X,1)="0" THEN PRINT "zero ";: GOTO 180
0170 N=VAL(MID$(F$,X,1)):GOSUB 9000
0180 NEXT X:GOTO 110
0200 GOSUB 9000:GOTO 110
9000 REM Rotina para imprimir os números
9010 IF N>1E9 THEN 9060
9020 IF N>1E6 THEN 9120
9030 IF N>1E3 THEN 9180
9040 IF N>99 THEN 9240
9050 GOTO 9300
9060 B=N:N=INT(N/1E9)
```

```
9070 GOSUB 9000
9080 PRINT " biliões";
9090 N=B-INT(B/1E9)*1E9
9100 IF N=0 THEN RETURN
9110 PRINT ", ";
9120 M=N:N=INT (N/1E6)
9130 GOSUB 9000
9140 PRINT " milhões";
9150 N=M-INT(M/1E6)*1E6
9160 IF N=0 THEN RETURN
9170 PRINT ", ";
9180 S=N:N=INT (N/1E3)
9190 GOSUB 9000
9200 PRINT " milhares";
9210 N=S-INT(S/1000)*1000
9220 IF N=0 THEN RETURN
9230 PRINT ", ";
9240 H=N.N=INT (N/100)
9250 GOSUB 9300
9260 PRINT " ";
9270 N=H-INT (H/100)*100
9280 IF N=0 THEN RETURN
9290 PRINT ", e ";
9300 RESTORE
9310 IF N>19 THEN 9420
9320 FOR D=1 TO N+1
9330 READ A$:NEXT D
9340 PRINT A$;" ";:IF N<21 THEN RETURN
9350 N=INT (N/10)*10
9360 IF N=0 THEN RETURN
9370 GOTO 9300
9420 FOR D=1 TO INT(N/10)+19
9430 READ A$:NEXT D
9440 PRINT A$;"e";:N=N-INT(N/10)*10
9450 IF N=0 THEN RETURN
9460 GOTO 9300
9465 DATA cem, duzentos, trezentos, quatrocentos, qui-
nhentos, seiscentos, setecentos, oitocentos, novecentos
9470 DATA um, dois, três, quatro, cinco, seis, sete, oito,
nove
```



```
9480 DATA dez, onze, doze, treze, catorze, quinze, dezas-
seis, dezassete, dezoito, dezanove
9490 DATA vinte, trinta, quarenta, cinquenta, sessenta, se-
tenta, oitenta, noventa
```

Exemplo de execução

```
Escreva um número?2
dois
Escreva um número? 34,1
trinta e quatro vírgula um
Escreva um número? 18
dezoito
Escreva um número? 123
cento e vinte e três
Escreva um número? 23456
vinte e três mil quatrocentos e cinquenta e seis
Escreva um número? 9.87654312E13
noventa e oito mil, setecentos e sessenta e cinco biliões quatro-
centos e trinta e um milhões, duzentos mil

READY
```

XVI

PARES (3,5 K)

Trata-se de uma aplicação a computador de um conhecido jogo de cartas em que estas são baralhadas e colocadas sobre a mesa viradas para baixo. Os jogadores podem virar duas cartas simultaneamente, e se forem um par o jogador retém-nas em seu poder e joga de novo. Se não são um par, são novamente viradas para baixo e o jogador seguinte continua. É um excelente teste de memória e um jogo agradável.

O programa utiliza uma matriz B para armazenar as cartas. Cada elemento contém o valor da carta, ou zero se o espaço correspondente estiver vazio.

Usei a função CHR$(X) bastante neste programa. É pena que a indústria de microcomputadores ainda não tenha estabelecido nenhum tipo de norma para a resposta a caracteres de impressão na extremidade inferior do código ASCII, porque esse facto obriga a fazer adaptações quando se transfere um programa de uma máquina para outra. No entanto, se conhecer bem a sua máquina, não deve ter dificuldades em traduzir as minhas instruções.

O programa começa no modo "scroll", se bem que pudesse igualmente iniciar-se em modo página. Utiliza-se um array A$ para guardar os nomes dos jogadores, sendo em seguida preenchida a matriz 6×6 com pares de valores, cuja posição é escolhida aleatoriamente. Se quiser, pode aumentar a dimensão desta matriz. A propósito, resolvi incluir o aviso da linha 220 porque os principiantes ficam normalmente ansiosos quando o computador não responde imediatamente.

O valor da carta é escolhido na linha 250, que produz um número aleatório na gama 65 a 80, ou o equivalente de A a Z no código ASCII. O posicionamento aleatório é feito pelas linhas 270 a 300.

Quando todos os espaços da matriz forem preenchidos, a linha 320 limpa o visor, coloca o terminal em modo página e desliga o cursor. A linha 1000 faz o retorno do cursor invisível, e em seguida é impressa a posição de todas as cartas, deixando um espaço em branco no caso de ter sido removida uma carta. A linha 1096 limpa o visor, perguntando-se em seguida ao jogador qual a carta que deseja virar.

O método usado para revelar a carta correcta consiste em fazer o retorno do cursor de tal modo que o ponto de partida seja conhecido, moven lo-o para baixo o número de linhas apropriado e depois fazendo-o mover-se ao longo do visor de modo a ficar directamente sobre o "+" impresso. Neste ponto imprimimos o código ASCII desse elemento da matriz. Como tudo isto é feito em cerca de um segundo, com o cursor desligado, o sinal + parece transformar-se magicamente numa letra (infelizmente não é possível mostrar isto na saída de impressora).

Se desejar (ou se for forçado a isso porque não consegue comutar a resposta do seu terminal) pode simplificar o programa um pouco omitindo todas as ordens CHR$.

Lista de variáveis

Q$	String de entradas
P	Número total de jogadores
A$(P)	Nomes dos jogadores
C9	Número de cartas ainda em jogo
B(X,Y)	Matriz das cartas
L	Contador de pares de cartas
W	Valor ASCII da carta
D	Contador geral
I	Número de linha da carta
J	Número de coluna da carta
C	Escolha do jogador (primeira ou segunda)
X	Escolha de linha pelo jogador
Y	Escolha de coluna pelo jogador
A1	Guarda o X da primeira carta
A2	Guarda o Y da primeira carta

C(X) Apresentação de duas cartas
S(X) Array das pontuações dos jogadores

Listagem do programa

```
0100 PRINT :PRINT
0110 PRINT TAB(15);"PARES"
0120 PRINT TAB(15)"———————"
0130 PRINT
0140 INPUT "Deseja instruções",Q$
0150 IF LEFT$(Q$,1) = "S" THEN GOSUB 9000
0160 P = 1:C9 = 36
0170 PRINT "NOME DO JOGADOR N.";P;" ";
0180 INPUT A$(P)
0190 IF A$(P) = ""THEN P = P−1:GOTO 210
0200 P = P + 1:GOTO 170
0210 PRINT :PRINT
0220 PRINT "Estou ocupado — já volto"
0230 DIM B (6,6)
0240 FOR L = 1 TO 18
0250 W = INT(RND(0)*26 + 65)
0260 FOR D = 1 TO 2
0270 I = INT(RND(0)*6 + 1)
0280 J = INT(RND(0)*6 + 1)
0290 IF B(I,J)<>0 THEN 270
0300 B(I,J) = W
0310 NEXT D:NEXT L
0320 PRINT CHR$(25); CHR$(25);CHR$(5);
0990 FOR P = 1 TO P
1000 PRINT CHR$(16);
1010 PRINT TAB(10);"eis as cartas — "
1020 PRINT
1030 PRINT "      1    2    3    4    5    6"
1040 FOR I = 1 TO 6:PRINT I;:FOR J = 1 TO 6
1050 IF B(I,J) 0 THEN PRINT "       + ";:GOTO 1070
1060 PRINT "       ";:GOTO 1070
1070 NEXT J:PRINT:NEXT I
1090 PRINT A$(P1);" —"
1095 FOR C = 1 TO 2
1096 PRINT CHR$(22);"        ";CHR$(5);
1100 INPUT "Número de linha, número de coluna",X,Y
1101 PRINT CHR$(5);IF C = 1 THEN A1 = X:A2 = Y:
GOTO 1110
1102 IF X<>A1 THEN 1110
1103 IF Y <>A2 THEN 1110
1104 PRINT CHR$(11);CHR$(22);
1105 PRINT "Acabou de ver essa carta!"
1106 PRINT CHR$(5);:GOTO 1100
1110 IF B(X,Y)<>0 THEN 1200
1120 PRINT CHR$(11);CHR$(22)
1130 PRINT "Essa carta já foi tirada":GOTO 1106
1190 REM — Revelar carta
1200 PRINT CHR$(16);
1210 FOR I = 1 TO X + 2
1220 PRINT CHR$(10);:NEXT I
1230 FOR J = 1 TO Y*4−2
1240 PRINT CHR$(09);:NEXT J
1250 PRINT "       ":CHR$(B(X,Y))
1255 C(C) = B(X,Y)
1260 PRINT CHR$(16)
1270 FOR I = 1 TO 10
1280 PRINT CHR$(10);:NEXT I
1290 NEXT C
1295 REM — Lance terminado. Um par?
1300 IF C(1) = C(2) THEN 1350
1305 FOR W = 1 TO 50:NEXT W:REM Pausa
1310 NEXT P1
1320 GOTO 990
1350 PRINT "UM PAR!!! FICA COM ";
1355 B(A1,A2) = 0:B(X,Y) = 0
1360 S(P1) = S(P1) + 1
1370 PRINT S(P1)
1380 C9 = C9−2:IF C9< = 0 THEN 2000
1390 PRINT "OUTRA VEZ, ";A$(P1);"!"
1400 GOTO 1000
2000 PRINT :PRINT
2010 FOR X = 1 TO 64:PRINT " = ";:NEXT X
2020 PRINT CHR$(14);:PRINT
```

```
2030 PRINT "PONTUAÇÕES FINAIS —"
2040 FOR X=1 TO P
2050 PRINT :PRINT A$(X);"      ";S(X)
2060 NEXT X:END
9000 PRINT "Este jogo ilustra um familiar jogo de cartas. ";
9010 PRINT "Vou apresentar-lhe 36 cartas viradas para baixo ";
9020 PRINT "podendo cada jogador virar duas cartas de cada";
9030 PRINT "vez. Se forem um par, são removidas pelo jogador ";
9040 PRINT "e este pode jogar de novo. O vencedor é aquele ";
9050 PRINT "que consegue maior número de pares — e aquele ";
9060 PRINT "que tem a melhor memória."
9070 PRINT "Para receber uma carta, escreva o número da ";
9080 PRINT "linha, uma vírgula, e o número da coluna."
9090 INPUT "Carregue em RETURN para começar...",Q$
9100 RETURN
```

Exemplo de execução

```
Deseja instruções? Não
NOME DO JOGADOR N. 1 JOÃO
NOME DO JOGADOR N. 2 MARIA
NOME DO JOGADOR N. 3 MANUEL
NOME DO JOGADOR N. 4

Estou ocupado — já volto
Eis as cartas —

JOÃO — Número de linha, número de colunas? 1,1

Número de linha, número de coluna? 1,2

Eis as cartas —
Número de linha, número de coluna? 6,4

Número de linha, número de coluna? 2,2
```



```
UM PAR!!! FICA COM 3
OUTRA VEZ, MANUEL!
Eis as cartas —

MANUEL — Número de linha, número de coluna? 5,1

Número de linha, número de coluna? 6,1

UM PAR!!! FICA COM 4
OUTRA VEZ, MANUEL!
Eis as cartas —

MANUEL —

UM PAR!!! FICA COM 9
OUTRA VEZ!
Eis as cartas —

MANUEL — Número de linha, número de coluna? 6,3

Número de linha, número de coluna? 6,5

UM PAR!!! FICA COM 10

= = = = = = = = = = = = = = = = = = = = = = = = = = =

JOÃO 1

MARIA 7

MANUEL 10
```

XVII

LEITURA RÁPIDA (5 K)

Encontramos agora um programa de concepção extremamente simples, mas que requer uma quantidade de memória razoável. De cada vez que o utilizador carrega na tecla RETURN, o visor é limpo e é apresentada uma frase durante alguns momentos, sendo apagada em seguida. Pede-se então ao utilizador que escreva aquilo que leu. Ao fim de 10 frases, o tempo durante o qual a frase é apresentada é ajustado para mais ou menos, sendo todo o processo repetido. Quando o utilizador resolve abandonar o programa, recebe uma pontuação de acordo com a sua velocidade de leitura.

A execução é de tal modo simples que não inclui nenhum exemplo no final da listagem. Este é o único programa do livro em que não o fiz, dado que o considerei desnecessário.

As frases escolhidas encontram-se evidentemente contidas em instruções DATA, podendo facilmente ser alteradas. O programa escolhe um número aleatório na gama 1 a 20, escolhendo e apresentando portanto uma das primeiras vinte frases. O tempo durante o qual a frase se mantém no visor é determinado pelo valor da variável T, que será alterado mais tarde. A linha 1020 limpa o visor e, juntamente com os espaços impressos pela linha 1050, coloca a frase num local bem visível do visor.

A linha 1070 conta muito simplesmente até ao valor previamente definido, e se o seu computador possui uma velocidade de execução maior ou menor do que o meu pode ajustar o valor de T, na linha 35. Tente apresentar a frase durante cerca de 1,5 segundos, depois do que será apagada pela linha 1080.

As linhas 1180 e 1185 consideram a possibilidade de o utilizador responder em maiúsculas, e se a comutação de um modo para outro for aborrecida as frases e palavras das linhas DATA podem ser todas escritas em maiúsculas.

A linha 1190 determina se a pontuação indica a necessidade de aumentar ou diminuir o tempo durante o qual cada frase se encontra visível, sendo T ajustado em função disso nas linhas 1200 ou 1220.

Quando se abandona o programa, a linha 9000 calcula a pontuação final a partir da última velocidade considerada, da última pontuação e de um pequeno valor aleatório — se bem que este possa ser omitido.

Além de constituir um óbvio teste de velocidade de leitura, o programa exige um pouco da nossa memória e do nosso conhecimento do teclado. Ilustra igualmente uma desvantagem da BASIC como linguagem de programação: a entrada deve ser exactamente igual. Consideremos por exemplo a frase "movimento contrário aos ponteiros do relórgio". Dever-se-á penalizar um utilizador que escreva dois espaços entre as palavras, ou se esqueça de um hífen entre palavras que o usem? A BASIC fá-lo-á certamente...

Lista de variáveis

T	Tempo
S	Pontuação
W	Palavra ou frase apresentada
Q$	String de entradas
X	Número aleatório
Y	Contador geral
T1	Contador de tempo
A$	Frase impressa
P	Indicador da frase
D	Pontuação

Listagem do programa

```
0010 PRINT TAB(12),"LEITURA RÁPIDA"
0020 PRINT TAB(12);"= = = = = = = = = ="
0030 PRINT
```

```
0035 T=40
0040 PRINT "Neste jogo, vou apresentar uma palavra ou frase ";
0050 PRINT "durante pouco tempo. Escreva em seguida o que ";
0060 PRINT "viu. Uma folha de papel e um lápis pode rão ";
0070 PRINT "ser-lhe úteis, mas por favor note que não terá ";
0080 PRINT "tempo suficiente para copiar tudo."
0090 PRINT
0110 PRINT "Por favor escreva em minúsculas. "
0120 INPUT "Carregue em RETURN para começar...",Q$
0150 PRINT :PRINT:PRINT "Obrigado. Vamos começar. "
0160 PRINT "Vou dar-lhe dez frases e depois indicar a sua ";
0170 PRINT " pontuação. "
0180 PRINT:PRINT "Observe com cuidado! Da próxima vez que ";
0190 PRINT "carregar a tecla RETURN, escrevo uma frase ";
0200 PRINT " exactamente a meio do visor. "
0210 PRINT "Cá vamos — "
0500 S=0
1000 FOR W=1 TO 10:INPUT "Carregue RETURN para começar...",Q$
1010 X=INT(RND(0)*20)+1:FOR Y=1 TO X:READ A$:NEXT Y
1020 PRINT CHR$(25);CHR$(25);PRINT:PRINT:PRINT:-PRINT
1050 PRINT "        ";A$
1060 REM — Temporização
1070 FOR T1=1 TO T::NEXT T1
1080 PRINT CHR$(25);CHR$(25)
1090 PRINT "Por favor escreva o que lê..."
1100 INPUT Q$
1110 IF Q$=A$ THEN 1140
1120 PRINT "Não está correcto."
1130 GOTO 1150
1140 PRINT "CORRECTO!":S=S+1
1150 NEXT W
1160 PRINT :PRINT "A sua pontuação foi ";S;" num máximo de 10."
```



```
1170 INPUT "Tente de novo",Q$
1180 IF LEFT$(Q$,1)="n" THEN 9000
1185 IF LEFT$(Q$,1)="N" THEN 9000
1190 IF S>5 THEN 1210
1195 PRINT "É melhor dar-lhe mais tempo para ler..."
1200 T=T+INT(T/3)
1205 RESTORE :P=0:S=0:GOTO 180
1210 PRINT "É melhor dar-lhe menos tempo para ler..."
1220 T=INT(T*6):GOTO 1205
1310 DATA a árvore grande, o meu velho gato, a casa branca
1315 DATA pobre gatinho, doi-lhe a pata
1320 DATA pobre velhota, cabelos brancos e ar cansado
1325 DATA três tordos, quatro grandes amigos, vinte sapos
1330 DATA vestidos de noite, comprido lápis verde
1335 DATA cinco sujeitos famosos, velha mala de couro, bilhete de comboio
1360 DATA um enorme rochedo cinzento, bonito espanador, erro de pilotagem
1365 DATA bilhetes de autocarro, cinzento esverdeado e oleoso
1370 DATA indígena selvagem, água caindo, queda de água
1375 DATA uma sandwich castanha, molho de salada, salada fria
1380 DATA sandwich de presunto, veleiro branco, ilhas do tesouro
1385 DATA sandwich de presunto, cabras montanhesas, três carneiros lanudos
1390 DATA velho casaco gasto, sobretudo coçado, cria de leoa
1395 DATA um horrível monstro, operários sujos, erva crescida
1405 DATA presunto cortado, bebé gritando, marinheiros no porto, carpinteiro naval
1420 DATA fatias de cebola, fantasmas rindo, aluno estúpido
1430 DATA dias ensolarados, folgas de material, pegas mal presas
1440 DATA más colheitas e ratos, relógio avariado, algumas pistas
1450 DATA na posição correcta, adivinhe a palavra misteriosa
1460 DATA um grande desafio, prisão de segurança
```

```
1470 DATA avaliar a resposta, patrulhada por guardas
1480 DATA movimento para trás, porcas, anilhas e parafusos
1490 DATA martelo e pratos, peixe quente e batatas
1500 DATA pão bem tostado e quente, prateleira rosa com
livros
1510 DATA movimento inverso, versão alfabética
1520 DATA estudos estratégicos, gestão económica
1530 DATA práticas contraproducentes, relógio avariado
1540 DATA melhorar a produção, acrobacia na televisão
1550 DATA computador estúpido, computador amigo
1560 DATA diferentes áreas de jogo, acrobacia na televisão
1570 DATA molho de tomate, torradas com manteiga,
vinagrete
1580 DATA mar, areia e diversão, terrenos diferentes
1590 DATA bater às portas, pobres e desamparados
1600 DATA crise internacional, velhas peúgas cinzentas
1610 DATA duplique a pontuação, perturbe um conde
1620 DATA retire nove números, melhore noventa bombas
1630 DATA símbolos químicos, retortas de laboratório
1640 DATA três pistas diferentes, três camiões que diferem
1650 DATA velocidade vertical decente, velocidade de
descida na vertical
1660 DATA imaginativo e exigente, exercícios de
investigação
1670 DATA medido diferentemente, diferente medida
1680 DATA escolha anual geral, tecnologia indiscutida
1690 DATA construção profissional, substituição protegida
1700 DATA publicação quinzenal, completa documentação
1710 DATA completamente revisto, extensivamente revisto
1720 DATA literatura educativa, designação semelhante
1730 DATA material copiado, análise de stocks
1740 DATA periféricos de computador, necessidade
apropriada
1750 DATA direitos pagos, disparate semântico
1760 DATA duplicado para máxima eficiência
1770 DATA material original pode ser reimpresso
1780 DATA virtualmente todos os fabricantes
1790 DATA estudante dedicado e ambicioso
1800 DATA frustrado e desinteressado
1810 DATA completamente revisto anualmente
```



```
1820 DATA imediato e autoritário
1830 DATA incompatibilidade de serviços
1840 DATA boletins periódicos de negócios
1850 DATA serviço inesperadamente eficiente
1860 DATA técnicas de gestão financeira
1870 DATA psicologicamente prejudicial
1880 DATA apagado transcendentalmente
1890 DATA acidentalmente dactilografado
1900 DATA suculenta e sensível feminista
1910 DATA empresa extraordinariamente dramática
1920 DATA aplicações e desenvolvimentos
1930 DATA gabinete creme agradável
9000 D=INT((5000/T)+(1000*S/T)+RND(0)*10))
9010 PRINT :PRINT
9020 PRINT "A sua pontuação final foi ";D
9030 PRINT "Adeus!"
9040 PRINT :PRINT
```

XVIII

TICKLE (3 K)

Este programa baseia-se num jogo comercial no qual as crianças se atam a si próprias fazendo nós ao mesmo tempo que tentam pôr as mãos, pés, etc., em marcas sobre um plástico. Esta adaptação foi sugerida por um amigo que encontrou um programa semelhante numa revista americana, de autor desconhecido. O jogo é bastante divertido para crianças quando existem números aproximadamente iguais de rapazes e raparigas, mas não há qualquer razão para "enganar" o programa transformando-o num jogo unisexo chamando a parte das raparigas rapazes e vice-versa. Não será necessário acrescentar que também pode ser divertido para adultos.

A linha 240 permite uma disparidade de um entre rapazes e raparigas; a "flag" F alterna entre -1 e + 1. Para além disto o programa é bastante simples.

Lista de variáveis

B	Número de rapazes
G	Número de raparigas
B$(X)	Array com nomes de rapazes
G$(X)	Array com nomes de raparigas
A$(X)	Array de partes activas
P$(X)	Array de partes passivas
F	Flag rapaz/rapariga

Listagem do programa

```
0100 DATA mão esquerda, mão direita, cotovelo esquerdo,
cotovelo direito
0110 DATA joelho esquerdo, joelho direito, pé esquerdo, pé
direito
0120 DATA ombro esquerdo, ombro direito, costas, cintura
0130 DATA perna esquerda, perna direita, braço esquerdo,
braço direito
0140 DATA anca esquerda, anca direita
0150 PRINT :PRINT:PRINT
0160 PRINT TAB(12);"TICKLE"
0170 PRINT TAB(12);"------------"
0180 PRINT
0190 PRINT "Este jogo deve ser jogado por um número ";
0200 PRINT "igual de rapazes e raparigas."
0210 PRINT
0220 INPUT "Quantos rapazes estão a jogar",B
0230 INPIRT "Quantas raparigas estão a jogar",G
0240 IF ABS(B-G)< = 1 THEN 0300
0250 PRINT "É NECESSÁRIO MAIS ";
0260 IF B-G<0 THEN PRINT "RAPAZES!":GOTO 0280
0270 PRINT "RAPARIGAS!"
0280 PRINT "VÁ BUSCÁ-LOS(AS) — EU ESPERO"
0290 GOTO 210
0300 PRINT "Por favor indique nomes dos rapazes —"
0310 FOR X = 1 TO B
0320 PRINT "Rapaz número ";X;" —";
0330 INPUT B$(X):NEXT X
0340 PRINT :PRINT:PRINT
0350 PRINT "obrigado. Agora os nomes";    0360 PRINT
"das raparigas —"
0370 FOR X = 1 TO G
0380 PRINT "Rapariga número ";X;" —"
0390 INPUT G$(X):NEXT X
0400 PRINT :PRINT
0410 PRINT "Obrigado."
0420 PRINT "É necessário um árbitro para este jogo."
0430 PRINT "  Vou indicar uma acção de cada vez  ";
0440 PRINT "sempre que o árbitro carregar em RETURN. ";
```

```
0450 PRINT " Os jogadores devem realizar essa acção ";
0460 PRINT "e manter cada posição até lhes ser dada ";
0470 PRINT " outra ordem. "
0480 PRINT "Vamos começar..."
0490 PRINT
0500 FOR X=1 TO 8
0510 READ A$(X)
0520 NEXT X
0530 FOR X=1 TO 10
0540 READ P$(X)
0550 NEXT X
0560 F=1
0600 PRINT "Por favor carregue em RETURN"
0610 PRINT
0620 INPUT Q$
0630 IF F=-1 THEN 800
0640 REM — escolher rapaz
0650 X=INT(RND(0)*B+1)
0660 REM — parte
0670 A=INT(RND(0)*8+1)
0680 REM — rapariga
0690 Y=INT(RND(0)*G+1)
0700 REM — parte
0710 P=INT(RND(0)*10+1)
0720 PRINT B$(X);" — Ponha ";A$(A);
0730 PRINT " sobre ";P$(P);" de ";G$(Y)
0740 F=F:PRINT:PRINT:GOTO 600
0800 REM — escolher rapariga activa
0810 Y=INT(RND(0)*G+1)
0820 A=INT(RND(0)*8+1)
0830 X=INT(RND(0)*B+1)
0840 P=INT(RND(0)*10)+1)
0850 PRINT :PRINT
0860 PRINT G$(Y);" — Ponha ";A$(A);" sobre ";
0870 PRINT P$(P);" de ";B$(X)
0880 F=-F:PRINT:PRINT:GOTO 600
```


Exemplo de execução

Este jogo deve ser jogado por um número igual de rapazes e raparigas.

Quantos rapazes estão a jogar? 2
Quantas raparigas estão a jogar? 3

Por favor indique nomes dos rapazes
Rapaz número 1 — João
Rapaz número 2 — Manuel

Obrigado. Agora os nomes das raparigas
Rapariga número 1 — Maria
Rapariga número 2 — Isabel
Rapariga número 3 — Ana

Obrigado.
É necessário um árbitro para este jogo.
Vou indicar uma acção de cada vez sempre que o árbitro carregar em RETURN. Os jogadores devem realizar essa acção e manter cada posição até lhes ser dada outra ordem.
Vou começar...

Por favor carregue em RETURN

João — Ponha cotovelo esquerdo sobre ombro esquerdo de Ana
Por favor carregue em RETURN

Isabel — Ponha mão esquerda sobre perna esquerda de Manuel
Por favor carregue em RETURN.

Manuel — Ponha joelho esquerdo sobre cintura de Isabel
Por favor carregue em RETURN

Ana — Ponha joelho direito sobre braço direito de Manuel
Por favor carregue em RETURN

João — Ponha pé esquerdo sobre ombro direiro de Ana
Por favor carregue em RETURN

Isabel — Ponha cotovelo direito sobre costas de Manuel
Por favor carregue em RETURN

Manuel — Ponha cotovelo direito sobre ombro esquerdo de Maria
Por favor carregue em RETURN

Maria — Ponha pé esquerdo sobre braço esquerdo de Manuel
Por favor carregue em RETURN

Manuel — Ponha pé esquerdo sobre ombro direito de Ana
Por favor carregue em RETURN

Isabel — Ponha cotovelo esquerdo sobre ombro direito de Manuel
Por favor carregue em RETURN

João — Ponha pé direito sobre anca esquerda de Isabel
Por favor carregue em RETURN

Ana — Ponha cotovelo direito sobre a anca esquerda de João
Por favor carregue em RETURN

João — Ponha joelho esquerdo sobre braço direito de Ana
Por favor carregue em RETURN

XIX

MENSAGENS SECRETAS (3 K)

É bastante interessante descodificar uma mensagem secreta — vê-la revelar-se letra a letra, salpicada de palavras estranhas aqui e ali. Este programa utiliza este interesse para apresentar 26 problemas ao utilizador, um para cada letra do alfabeto. Pode-se complicá-lo ainda mais, de tal modo que quando a mensagem seja finalmente descodificada uma outra dirija o utilizador para um prémio escondido.

A mensagem secreta é guardada nas instruções DATA das linhas 60 a 80, podendo ser facilmente alterada à medida que os utilizadores a conseguem descobrir. Se alterar a mensagem, note que cada linha é guardada numa única variável de string, não devendo portanto ter um comprimento superior ao definido pela sua versão da BASIC. Note ainda que o número de linhas de dados pode ter de ser alterado na linha 50. No interior da mensagem secreta só são permitidas letras e pontos finais — números, pontos de interrogação, etc., não podem ser usados.

As linhas 100 a 170 contêm as 26 chaves e as respectivas respostas. Se bem que tenha apresentado apenas cidades, é possível substituí-las por qualquer outra coisa. Note uma vez mais que nenhuma pergunta ou resposta pode ser mais comprida do que o string que a contém. Se por acaso incluir alguma coisa excessivamente longa, a BASIC truncá-la-á. É obviamente importante que existam exactamente 26 perguntas e respostas.

A fim de poder inserir cada letra por sua vez na mensagem secreta, o utilizador deve fornecer a resposta correcta à pergunta associada a essa letra. Com alguma malícia, o leitor pode associar as perguntas mais difíceis às letras mais comuns...

Um possível melhoramento consistiria em levar o programa a ter acesso a um conjunto diferente de mensagens e perguntas em disco, se dispuser deste periférico.

As strings C$(1) a C$(3) contêm a mensagem parcialmente descodificada. Ajustei o comprimento das strings para 50 caracteres (na linha 50) a fim de cada linha de mensagem caber numa string. Se não puder fazer isto, terá de dividir as linhas de mensagem com separadores (provavelmente pontos e vírgulas) e ajustar a variável L em função disso.

O array A regista as letras que foram descodificadas. Cada elemento inicia-se pelo valor 0, transformando-se em 9 quando a chave respectiva é descoberta.

Os programadores principiantes devem observar as funções de string que surgem a partir da linha 220. Nesta linha observa-se separadamente cada caracter de T$, inserindo-se em D$ para exame. Se se trata de um espaço ou um ponto final, é simplesmente acrescentado no final da string. Se não é, ou seja, se se trata de uma letra, é necessário verificar o array A. O código ASCII do caracter variará entre 65 para A e 90 para Z; subtraindo a ele 64 obtemos uma variação entre 1 e 26, equivalente aos 26 elementos do array de comparação. Se o conteúdo do elemento não é zero, a chave já foi resolvida e portanto o caracter é acrescentado a C$(X); senão acrescenta-se um ponto.

O leitor deve ser capaz agora de compreender o funcionamento das linhas 340 e 360.

Lista de variáveis

Q$	String de entradas
L	Número de linhas DATA na mensagem
A(X)	Array correspondente a cada letra
X	Contador
C$(X)	Linhas da mensagem descodificadas ou parcialmente descodificadas
T$	String de transferência
Y	Contador
D$	Um caracter único de T$
Q	Número de letra de Q$, 1-26
R$	Pergunta ou chave
S$	Resposta correcta
Z	Contagem de chaves resolvidas

Listagem do programa

```
0010 PRINT "MENSAGENS SECRETAS"
0020 PRINT "= = = = = = = = = = = = ="
0030 INPUT "Deseja instruções",Q$
0040 IF LEFT$(Q$,1)="S" GOSUB 1000
0050 L=3:STRING=50
0060 DATA APARECE À ENTRADA DO ZOO ANTES
DE
0070 DATA   NASCER O SOL. TRAZ UMA ARMA. O
0080 DATA   JOÃO SUSPEITA DE NÓS. MANUEL.
0090 DATA EUA, WASHINGTON, ESCÓCIA,
EDINBURGH, FRANÇA, PARIS
0100 DATA ALEMANHA OCIDENTAL, BONA,
CANADÁ, OTTAWA, QUÉNIA, NAIROBI
0110 DATA RÚSSIA, MOSCOVO, FINLÂNDIA,
HELSÍNQUIA, TURQUIA, ISTAMBUL
0120 DATA INGLATERRA, LONDRES, IRLANDA,
DUBLIN, CHINA, PEQUIM
0130 DATA BRASIL, BRASÍLIA, SUÉCIA,
ESTOCOLMO, ROMÉNIA, BUCARESTE
0140 DATA GALES, CARDIFF, JAPÃO, TÓQUIO,
AUSTRÁLIA, CAMBERRA
0150 DATA AFEGANISTÃO, CABUL, VENEZUELA,
CARACAS, PERÚ, LIMA
0160 DATA MÉXICO, CIDADE DO MÉDIXO, ITÁLIA,
ROMA, ESPANHA, MADRID
0170 DATA COLÔMBIA, BOGOTÁ, BÉLGICA,
BRUXELAS
0180 DIM A(26)
0190 REM — Actualizar mensagem descodificada
0200 RESTORE ;FOR X=1 TO L:C$(X)=""
0210 READ T$
0220 FOR Y=1 TO LEN(T$):D$=MID$(T$,Y,1)
0230 IF D$="." THEN C$(X)=C$(X)+D$:GOTO 270
0240 IF D$=" " THEN C$(X)=C$(X)+D$:GOTO 270
0250 IF  A(ASC(D$)-64)=0  THEN  C$(X)=C$(X)+".":
GOTO 270
0260 C$(X)=C$(X)+MID$(T$,Y,1)
0270 NEXT Y:NEXT X
```

```
0280 REM — Imprimir forma codificada
0290 PRINT :FOR X=1 TO L:PRINT C$(X):NEXT X
0300 REM — Procurar e verificar solução
0310 PRINT
0320 INPUT "Que letra quer experimentar ",Q$
0330 Q=ASC(Q$)-64:IF Q<=0 THEN 350
0340 IF Q<=26 THEN 360
0350 PRINT "Por favor escreva uma letra (A-Z)":GOTO
320
0360 IF A(Q)<>0 PRINT "Essa chave já foi resolvida":
GOTO 320
0370 REM Descobrir e imprimir pergunta
0380 RESTORE :FOR X=1 TO L:READ R$:NEXT X
0390 FOR X=1 TO Q:READ R$,S$:NEXT X
0400 PRINT "Qual é a capital de ";R$;
0410 INPUT A$:IF A$<>S$ PRINT "É pena — está
errado":GOTO 320
0420 PRINT "CERTO! RESOLVEU UMA CHAVE"
0430 A(Q)=9:Z=Z+1:IF Z<>26 THEN 200
0440 PRINT :PRINT
0450 PRINT "ÓPTIMO! RESOLVEU TODAS AS
CHAVES!"
0460 PRINT "Se conseguir dizer ao paizinho qual é a ";
0470 PRINT " capital do Botswana, ele dá-lhe um prémio!"
0480 END
1000 PRINT "Este jogo é divertido. Possuo uma mensagem ";
1010 PRINT " secreta para você resolver. Vou revelar ";
1020 PRINT "uma letra de cada vez se resolver as chaves ";
1030 PRINT " que antes lhe apresento. ";
1040 PRINT "Por exemplo, apresento-lhe toda as letras
E's ";
0150 PRINT "se resolver uma chave, todas as A's se resolver ";
1060 PRINT "outra, etc."
1070 INPUT "Carregue em RETURN para começar...",Q$
1080 RETURN
```


Exemplo de execução

```
Deseja instruções? Não
....... . ....... .. ... .....  ..
...... . .... .... ... ..... .
.... ........ .. .... .......

Que letra quer experimentar? A
Qual é a capital de EUA? NEW YORK
É pena — está errado
Que letra quer experimentar? E
Qual é a capital de CANADÁ? OTTAWA
CERTO! RESOLVEU UMA CHAVE

....E.E . E...... .. ... ...E. .E
....E. . .... .... ... ..... .
.... ....E... .E .... ....E..

Que letra quer experimentar? S
Qual é a capital de AFEGANISTÃO? CABUL
CERTO! RESOLVEU UMA CHAVE

....E.E . E...... .. ... ...ES .E
..S.E. . S... .... ... ..... .
.... S.S.E... .E ..S. ....E..

A.ARECE A ENTRADA DO .00 ANTES DE NASCER O
SOL. TRA. UMA ARMA. O JOÃO SUS.EITA DE NOS.
MANUEL.

Que letra quer experimentar? P
Qual é a capital de GALES? CARDIFF
CERTO! RESOLVEU UMA CHAVE
APARECE A ENTRADA DO .00 ANTES DE NASCER O
SOL. TRA. UMA ARMA. O JOÃO SUSPEITA DE NOS.
MANUEL.

Que letra quer experimentar? Z
Qual é a capital de BÉLGICA? BRUXELAS
CERTO! RESOLVEU UMA CHAVE
```

APARECE A ENTRADA DO ZOO ANTES DE NASCER O SOL. TRAZ UMA ARMA. O JOÃO SUSPEITA DE NOS.
MANUEL

Que letra quer experimentar? V
Qual é a capital de MÉXICO? CIDADE DO MÉXICO
CERTO! RESOLVEU UMA CHAVE

ÓPTIMO! RESOLVEU TODAS AS CHAVES!
Se conseguir dizer ao paizinho qual é a capital do Botswana, ele dá-lhe um prémio!

READY

XX

BARCO A MOTOR (3 K)

Neste jogo é necessário conduzir um barco num lago, evitando obstáculos e as margens. O lago é computadorizado, e portanto é quadrado... As bóias (obstáculos) são dispostas de modo diferente para cada jogo.

No início, o barco encontra-se no canto superior esquerdo, sendo a sua posição marcada por um B; e parte sempre em direcção sudeste, aproximando-se do meio do lago. Pode conduzir o barco, mas só pode pará-lo encalhando-o na costa — o que custa pontos — ou ficando sem combustível ao fim de 100 movimentos. Pode também perder pontos no caso de entrar em contacto com uma bóia.

A condução do barco merece ser estudada em detalhe. O rumo varia de 0 a 7, sendo 0 equivalente a Norte e prosseguindo em intervalos de 45 graus até 7, noroeste. Quando o leme é virado para bombordo ou estibordo, mantém-se nessa posição até ser novamente endireitado ou rodado no sentido contrário. O barco move-se então em círculo; mas a fim de evitar que o leitor engane o computador e se mova em círculo os 100 movimentos, o programa não permitirá mais do que sete movimentos sem mudar de rumo, o que equivale aproximadamente a 360 graus.

Em cada movimento o utilizador deve escrever uma ou várias ordens, indicadas no quadro ao lado da listagem. A string de entrada é examinada nas linhas 241 a 249, e os puristas não apreciarão o método; existem certamente outros melhores. No entanto, os principiantes em programação compreendem-no certamente melhor, e os que já tiverem experiência saberão escrever estas linhas de outro modo.

Um aviso: ao jogar, não use mais mapas do que os necessários, pois cada um custa-lhe 15 pontos.

Devo agradecer a "Personal Computer World" a autorização para usar este jogo, que foi publicado pela primeira vez em 1979.

Lista de variáveis

D1	Posição actual, linha
D2	Posição actual, coluna
D3	Rumo (1-7)
D4	Leme (-1,0, + 1)
A(X,Y))	Matriz do lago, onde 0 = espaço 2 = percurso 1 = barco 3 = bóia
X	Contador de linhas
Y	Contador de colunas
Z	Número aleatório 0-9
T	Pontuação actual
M	Contador de movimentos
H1	Movimentos com o leme actual
H	Posição do leme
E	Incremento para norte
F	Incremento para leste

Listagem do programa

```
0010 PRINT "BARCO A MOTOR"
0020 PRINT "Partida..."
0030 D1 = 1
0040 D2 = 1
0050 D3 = 3
0060 D4 = 0
0070 DIM A(12,14)
0080 FOR X = 1 TO 12
0090 FOR Y = 1 TO 14
0100 Z = RND(0)*10
0110 IF Z<9 THEN 140
0120 A(X,Y) = 3
0130 GOTO 150
0140 A(X,Y) = 0
```



```
0150 NEXT Y
0160 NEXT X
0170 A(1,1) = 1
0180 T = 0
0190 M = 0
0210 GOSUB 1000
0220 H1 = 0
0230 PRINT "ORDEM ";
0235 A(D1,D2) = 1
0240 INPUT C$:C$ = LEFT$(C$,1)
0241 IF C$ = "0" THEN 550
0242 IF C$ = "M" THEN 280
0243 IF C$ = "H" THEN 320
0244 IF C$ = "S" THEN 440
0245 IF C$ = "E" THEN 530
0246 PRINT "ORDEM NÃO ENTENDIDA":GOTO 240
0280 REM mapa
0290 T = T-15
0300 GOSUB 1000
0310 GOTO 230
0320 REM leme
0330 INPUT "LEME",C$
0332 C$ = LEFT$(C$,1):IF C$ = "L" THEN H = -1:
GOTO 350
0334 IF C$ = "R" THEN H-1:GOTO 350
0336 IF C$ = "C" THEN H = 0:GOTO 350
0338 PRINT "NÃO COMPREENDIDO":GOTO 330
0350 IF H<>H1 THEN 400
0360 H1 = H1 + 1
0370 IF H1< = 7 THEN 400
0380 PRINT "NÃO É PERMITIDO RODAR"
0385 PRINT "EM CÍRCULOS"
0390 GOTO 33
0400 H1 = 1
0410 D4 = H
0420 GOTO 550
0430 REM Informação
0440 T = T-5
0450 PRINT "POSIÇÃO: ";D1;",";D2
0460 PRINT "RUMO: ";D3
```

```
0470 PRINT "LEME: ";
0480 IF D4=-1 THEN PRINT "BOMBORDO"
0490 IF D4=0 THEN PRINT "DIREITO"
0500 IF D4=1 THEN PRINT "ESTIBORDO"
0510 PRINT "PONTUAÇÃO: ";T
0520 GOTO 230
0530 PRINT "POSIÇÃO FINAL ";D1;",";D2
0540 GOTO 740
0550 REM movimento
0560 DATA -1, 0,-1, × 1,0,1,1,1,1,0,1,-1,0,-1,-1,-1
0570 D3=D3+D4
0580 IF D3>7 THEN LET D3=0
0590 IF D3<0 THEN LET D3=7
0600 FOR I=1 TO D3+1
0610 READ E, F
0620  NEXT I
0625 RESTORE
0630 A(D1,D2)=2
0640 D1=D1+E
0650 D2=D2+F
0652 IF D1<C1 THEN 720
0654 IF D1>12 THEN 720
0656 IF D2<1 THEN 720
0658 IF D2>14 THEN 720
0660 IF A(D1,D2)=3 THEN 690
0670  T=T+10
0674 M=M+1
0676 IF M>=100 THEN 800
0680 GOTO 230
0690  PRINT "COLISÃO EM ";D1;",";D2
0700 T=T-50
0704 D1=D1-E
0708 D2=D2-F
0710 GOTO 230
0720 PRINT "ENCALHOU!"
0730 T=T-100
0740 PRINT "A SUA PONTUAÇÃO FOI ";T
0750 PRINT
0755 INPUT "QUER VER O MAPA FINAL",Q$
0756 IF LEFT$(Q$,1)="N" THEN 760
```



```
0757 GOSUB 1000
0760  PRINT "ESCREVA RUN PARA JOGAR DE
NOVO";
0770 END
0800 PRINT "ÓPTIMO! TERMINOU A CORRIDA"
0810 PRINT
0820 T=T+100
0830 GOTO 740
1000 REM mapa
1010 PRINT
1020 PRINT "      1 2 3 4 5 6 7 8 9 0 1 2 3 4"
1030 FOR X=1 TO 12
1040 PRINT X;
1050 IF X<10 THEN PRINT " ";
1060 FOR Y=1 TO 14
1070 IF A(X,Y)=0 THEN PRINT " .";
1080 IF A(X,Y)=1 THEN PRINT " B";
1090 IF A(X,Y)=3 THEN PRINT " 0";
1100 IF A(X,Y)=2 THEN PRINT " *";
1110 NEXT Y
1120 PRINT
1130 NEXT X
1140 PRINT "RUMO: ";D3
1150 RETURN
```

As ordens neste programa são as seguintes
0 — nenhuma
M — mapa
S — pontuação e posição
E — fim de jogo
H — leme (com os comandos:)
  L — esquerda
  R — direita
  C — central

Exemplo de execução

```
Partida...

RUMO: 3
Deseja instruções? 0
ORDEM? 0
ORDEM? 0
ORDEM? 0
ORDEM? 0
ORDEM? 0
ORDEM? 0
ORDEM? M

RUMO: 3
ORDEM? 0
ORDEM? 0
ORDEM? R
ORDEM? 0
ORDEM? 0
ORDEM? 0
ORDEM? 0
ORDEM? C


ORDEM? 0
ORDEM? M


RUMO: 0
ORDEM? 0
ORDEM? Q
ORDEM NÃO COMPREENDIDA
0
ORDEM? 0
ORDEM? R
ORDEM? 0
ORDEM? C
ORDEM? S
POSIÇÃO: 4 ,9
RUMO: 2
LEME : DIREITO
```



```
PONTUAÇÃO: 165
ORDEM? 0
ORDEM? 0
ORDEM? 0
ORDEM? L
ORDEM? 0
ORDEM? 0
ORDEM? 0
ORDEM? C
ORDEM? S
POSIÇÃO: 1 ,10
RUMO:6

LEME: DIREITO
PONTUAÇÃO: 240
ORDEM? MAPA


RUMO: 6
ORDEM? R
ENCALHOU!
A SUA PONTUAÇÃO FOI 127
QUER VER O MAPA FINAL? NÃO
ESCREVA RUN PARA JOGAR DE NOVO
READY
```

XXI

CAÇA A SUBMARINOS (5 K)

Como o título implica, o objectivo deste jogo consiste em descobrir e eliminar um submarino inimigo submerso, mas infelizmente este não está passivo! O inimigo move-se a metade da velocidade do navio de superfície e pode fazer manobras inesperadas. Mais ainda, se consegue atingir o rebordo da área de jogo, o leitor perde! Perde também se tentar sair da área de jogo, apesar de se considerar que está em mar aberto, ou se gastar todas as suas cargas de profundidade.

Como o inimigo está submerso, é óbvio que o leitor não o vê. De facto, a única maneira de avaliar a sua posição consiste em utilizar o seu ASDIC, instrumento que lhe indica a direcção polar relativamente à sua posição actual. Mas resta-lhe uma consolação: tal como acontece com o navio de superfície o submarino também não pode inverter instantaneamente o seu movimento, podendo no entanto alterar o rumo de 45 graus em cada movimento.

Os rumos são indicados como no jogo anterior, aplicando-se a mesma regra quanto ao uso do leme. Neste jogo, no entanto, o seu navio pode rodar em círculos à sua vontade — mas o submarino está provavelmente a afastar-se entretanto...

O programa utiliza uma forma alternativa de descodificar ordens (nos programas seguintes apresentar-se-ão outras). Este método requer que o utilizador indique o número correspondente à sua ordem, como no menú de «Aritmética», mas sem poder dispor da lista de opções. Se utilizar este método, verificará que aprende rapidamente as ordens, podendo entretanto recorrer a uma lista em papel.

Os incrementos usados neste jogo e no anterior são obtidos a partir das instruções DATA na linha 560. Se as contar verificará

que são 16, sendo lidas aos pares (linha 600), conforme o rumo. Por exemplo, o rumo 0 provoca uma transferência do primeiro par, etc. O rumo 0 equivale a Norte, e o par de incrementos é -1 e 0. Aplicando à posição actual, a fiada diminui de 1 e a posição lateral (direita-esquerda) não é alterada. Quaisquer outras direcções são tratadas do mesmo modo.

Lista de variáveis

Q$	String de entradas
D	Cargas de profundidade restantes
D1	Posição do contratorpedeiro — linha
D2	Posição do contratorpedeiro — coluna
D3	Rumo do contratorpedeiro
D4	Leme do contratorpedeiro
D6	Flag indicadora do lançamento de cargas de profundidade
A(X,Y)	Matriz da área de jogo, onde 0 = mar 2 = percurso 1 = contratorpedeiro 4 = rumo do submarino
X	Contador de linhas
Y	Contador de colunas
S1	Posição do submarino — linha
S2	Posição do submarino — coluna
S3	Rumo do submarino
S4	Leme do submarino
B	Contador de movimentos
C	Número da ordem
H	Leme desejado
I	Contador geral
E	Incremento da posição — norte/sul
F	Incremento da posição — leste/oeste

Listagem do programa

```
0010 PRINT "CAÇA A SUBMARINOS"
0020 INPUT "Deseja instruções (s-n)", Q$
0030 IF LEFT$(Q$,1) = "S" THEN GOSUB 2000
```

```
0040 PRINT "Partida..."
0045 REM — situação inicial
0050 D1=1:D2=1:REM — coordenadas do contratorpedeiro
0060 D3=3:REM — rumo
0070 D6=0:D4=0
0080 DIM A (12,14)
0085 REM — limpar oceano
0090 FOR X=1 TO 12:FOR Y=1 TO 14
0100 A (X,Y)=0:NEXT Y:NEST X
0130 S3=INT (RND (0) * 7):REM — rumo inicial do submarino
0150 S4=0:REM — leme do submarino
0160 A (1,1)=1:REM — marcar posição do contratorpedeiro
0170 REM — marcar posição do submarino
0180 S1=INT (RND (0) * 12+1):S2=INT (RND (0) *
14+1):A (S1, S2)=3
0210 D=24:REM cargas de profundidade
0220 B=1:REM contagem dos movimentos do contratorpedeiro
0230 PRINT: INPUT "Ordem", C
0250 C=INT (ABS (C) ):IF C>7 THEN 230
0270 ON C+1 GOTO 540,280,320,430,530,800,950,990
0280 REM — display
0300 GOSUB 1000:GOTO 230
0320 REM — Mudança de leme
0330 INPUT "Mudança de leme", H
0350 IF ABS (H)>1 PRINT "Não comprendo.":GOTO 330
0385 PRINT "Sim, sr. Comandante,":D4=H:GOTO 540
0430 REM — Pedido de informação
0450 PRINT "Posição: ";D1;",";D2
0460 PRINT "Rumo: ";D3
0470 PRINT "Leme: ";
0480 IF D4=-1 PRINT "Bombordo"
0490 IF D4=0 PRINT "Direito"
0500 IF D4=1 PRINT "Estibordo"
0520 GOTO 230
0530 PRINT "Fora do mapa em";D1;",";D2:GOTO 730
0535 REM — mover contratorpedeiro
0540 IF D6<>1 THEN 570
0550 D=D-2:IF D<0 PRINT "Gastas todas as cargas":
GOTO 730
0560 DATA -1,0,-1,1,0,1,1,1,1,0,1,-1,0,-1,-1,-1
```



```
0570 D3=D3+D4:REM — Ajustar rumo orientando leme
0580 IF D3>7 THEN D3=0
0590 IF D3<0 THEN D3=7
0595 REM — Ajustar coordenadas
0600 FOR I=1 TO D3+1:READ E,F:NEXT I:RESTORE
0605 A(D1,D2)=2:REM — Memorizar movimentos
0610 D1=D1+E:D2=D2+F:REM — Nova posição
0615 IF D1<1 THEN 720
0620 IF D1>12 THEN 720
0625 IF D2<1 THEN 720
0630 IF D2>14 THEN 720
0635 A(D1,D2)=1:D5=D5-2: REM — Largar cargas
0636 REM — Verificar se atingem
0640 IF D1<>S1 THEN 710
0645 IF D2<>S2 THEN 710
0650 IF D6<>1 THEN 710:REM — Flag das cargas
0660 PRINT "Carga de profundidade atinge submarino!
Ganhou!":GOTO 750
0700 REM — Contratorpedeiro move-se duas vezes por cada
vez do sub.
0710 B=B+1:IF B<=2 THEN 460
0715 GOTO 1160:REM — Movimento do submarino
0720 PRINT "Contacto perdido."
0730 PRINT " Perdeu."
0750 PRINT:INPUT "Deseja um último mapa (s-n)",Q$
0760 IF Q$="S" COSUB 1000
0770 END
0800 REM — Relatório ASDIC
0810 IF ABS (D1-S1)>5 THEN 930
0820 IF ABS (D2-S2)>5 THEN 930
0830 PRINT "Eco na direcção";
0840 IF D1<S1 AND D2>S2 PRINT "Sudoeste"
0842 IF D1<S1 AND D2<S2 PRINT "Sudeste"
0844 IF D1>S1 AND D2>S2 PRINT "Noroeste"
0846 IF D1>S1 AND D2<S2 PRINT "Nordeste"
0848 IF D1>S1 PRINT "Norte"
0850 IF D1<S1 PRINT "Sul"
0860 IF D2>S2 PRINT "Oeste"
0870 IF D2<S2 PRINT "Leste"
0880 IF D2<>S2 THEN 230
```

```
0890 IF D1<>S1 THEN 230
0900 PRINT "Por baixo de nós!"
0910 IF D1<>S1 THEN 230
0920 IF D6=1 THEN 660
0925 GOTO 230
0930 PRINT "Ausência de eco":GOTO 230
0950 REM — Cargas de profundidade
0960 PRINT "Lançar cargas!":D6=1:GOTO 230
0990 PRINT "Cessar fogo!":D6=0:GOTO 230
1000 REM — display
1010 PRINT
1020 PRINT "    12345678901234"
1030 FOR X=1 TO 12:PRINT X;
1050 IF X<10 PRINT "";
1060 FOR Y=1 TO 14
1070 IF A(X,Y)=0 PRINT " .";
1080 IF A(X,Y)=1 THEN PRINT " D";
1090 IF A(X,Y)=2 THEN PRINT " *";
1105 IF A(X,Y)=4 THEN PRINT " +";
1110 NEXT Y:PRINT:NEXT X
1140 PRINT "Rumo: ",D3:RETURN
1160 REM — Movimentos do submarino
1170 X=RND(0)*10
1180 IF X>7 THEN S4=-1
1190 IF X<3 THEN S4=1
1200 S3=S3+S4
1210 IF S3>7 THEN S3=0
1220 IF S3<0 THEN S3=7
1230 FOR I=1 TO S3+1:READ E,F:NEXT I:RESTORE
1270 A(S1,S2)=4:S1=S1+E:S2=S2+F
1300 IF S1<1 THEN 1350
1310 IF S1>12 THEN 1350
1320 IF S2<1 THEN 1350
1330 IF S2>14 THEN 1350
1340 B=1:GOTO 460
1350 PRINT "Submarino escapa!":GOTO 730
2000 PRINT "Você é o Comandante de um contratorpedeiro
que ";
2010 PRINT " persegue um submarino inimigo. Pode escutar
o ";
```



```
2020 PRINT " ruído deste com equipamento ASDIC, e tentar ";
2030 PRINT " destruí-lo com cargas de profundidade, mas o ";
2040 PRINT " alcance do ASDIC é limitado e o submarino es
tá ";
2050 PRINT "em movimento. Você perde se o submarino fugir
";
2055 PRINT " da área do mapa, ou se ficar sem cargas de ";
2060 PRINT "profundidade. Possui 24 cargas, e deve lançá-
-las ";
2070 PRINT " aos pares. "
2080 INPUT "Por favor carregue em RETURN.",Q$
2090 PRINT "Ordens permitidas: "
2100 PRINT " 0 — Manter leme actual."
2110 PRINT " 1 — Imprimir mapa."
2120 PRINT " 2 — Mudar o leme."
2130 PRINT " 3 — Pedido de informação."
2140 PRINT " 4 — Terminar jogo."
2150 PRINT " 5 — Relatório ASDIC."
2160 PRINT " 6 — Iniciar lançamento das cargas."
2170 PRINT " 7 — Terminar lançamento de cargas."
2180 PRINT
2190 PRINT "Leme — 0=Direito, 1=Estibordo, 2=Bombor
do."
2210 PRINT "Os Rumos são dados por 0-7, sendo 0 Norte e ";
2220 PRINT " aumentando no sentido dos ponteiros do
relógio em ";
2230 PRINT "passos de 54 graus. Sudoeste é portanto rumo
5."
2240 INPUT "Por favor carregue em RETURN",Q$
2250 RETURN
```

Exemplo de execução

```
Deseja instruções? N
Partida...
Ordem? 5
Ausência de eco
```

Ordem? 0
Rumo: 3
Leme: Direito

Ordem? 5
Ausência de eco

Ordem? 0
Rumo: 3
Leme: direito

Ordem? 5
Eco na direcção sudoeste

Ordem? 2
Mudança de leme? 1
Sim, sr. Comandante
Rumo: 4
Leme: Estibordo

Ordem? 5
Eco na direcção sudoeste

Ordem? 0
Submarino escapa!
Perdeu.

N
CAÇA A SUBMARINOS
Deseja instruções? N
Partida...

Ordem? 5
Ausência de eco

Ordem? 0
Rumo: 3
Leme: Direito

Ordem? 5
Eco na direcção sudeste

Ordem? 0
Rumo: 3
Leme: Direito

Ordem? 5
Eco na direcção sudeste

Ordem? 0
Rumo: 3
Leme: Direito

Ordem? 5
Eco na direcção sudeste

Ordem? 6
Lançar cargas!

Ordem? 5
Eco na direcção sudeste

Ordem? 0
Rumo: 3
Leme: Direito

Ordem? 5
Eco na direcção sudoeste

Ordem? 2
Mudança de leme? 1
Sim, sr. Comandante.
Rumo: 4
Leme: Estibordo

Ordem? 5
Eco na direcção sudoeste

Ordem? 0
Rumo: 5
Leme: Estibordo

Ordem? 5
Eco na direcção sul

Ordem? 2
Mudança de leme? -1
Sim, sr. Comandante
Rumo: 4
Leme: Bombordo

Ordem? 5
Eco na direcção sul

Ordem? 2
Mudança de leme? 0
Sim, sr. Comandante
Carga de profundidade atinge submarino! Ganhou!

Deseja um último mapa (s-n)? S

XXII

LEILÃO (5,5 K)

Não sei bem porque razão este jogo é tão fascinante. Talvez seja a esperança de obter uma pechincha, apesar de precária; talvez se trate de o programa conseguir captar o ambiente da sala de leilões, ou talvez ainda o prazer intelectual de compreender as pistas lançadas pelos outros compradores. Mais provavelmente, no entanto, será a possibilidade de vencer os outros jogadores. Qualquer que seja a razão, este jogo é considerado o melhor entre as pessoas da minha família, novas ou de maior idade, e estou certo de que encontrará muitos outros defensores.

O jogo é suficientemente bem explicado na listagem, e a sintaxe do programa não é de modo algum difícil; mas a táctica a seguir merece certamente algum estudo.

Depois de o jogo ter começado, apresentam-se lotes a leilão em grupos de vinte artigos, com um descanso "para tomar café" depois de cada lote. O jogo só pode ser terminado num destes intervalos, o que permite não dar a qualquer jogador uma vantagem em relação aos outros.

Os jogadores são designados pelo nome, actuando o computador como leiloeiro e contabilista. Os jogadores estão portanto a competir uns com os outros, não com o computador. Existe no entanto um pormenor importante, que assume a forma de um (por vezes dois) licitadores secretos. Estes representam aquelas pessoas verdadeiramente conhecedoras que muitas vezes são antiquários profissionais. Conhecem o verdadeiro valor dos objectos antigos e, ao vê-los fazer as suas ofertas relativas a um dado artigo, os jogadores podem ter uma ideia do verdadeiro valor deste. Essas pessoas podem no entanto cometer erros. Por vezes sobrestimam o valor de um artigo ou não apreciam devidamente

algo bastante valioso — e é então que o jogador mais aventuroso pode fazer a sua oferta.

Começando com um capital de $100 cada, os jogadores não podem certamente fazer muitas ofertas, mas podem adquirir bons artigos e aumentar o seu capital.

Para ter êxito neste jogo, portanto, é necessário possuir-se uma mistura de precaução, espírito de aventura, experiência do jogo e sorte. Talvez isto explique a sua popularidade.

Depois de definir um array P$(X) contendo os nomes ou pseudónimos de todos os jogadores, e uma matriz contendo o dinheiro de que dispõem, é posto à venda o primeiro lote com um valor definido pela linha 650. O leitor notará que utilizo três números aleatórios entre 0 e 0,99999, multiplicando-os uns pelos outros — uma técnica usada por vezes a fim de dar maior peso à parte inferior da gama de valores considerada. Multiplicando por 10 000 consegue-se preços relativamente baratos, apenas alguns deles atingindo os $10 000.

Determina-se em seguida o momento em que o licitador secreto entra em jogo, se é que o faz. Se um jogador não oferece nada, a sua matriz é carregada com o valor -9, e o leiloeiro não o incomodará mais durante a venda do lote presente. No caso contrário, o valor da sua oferta é colocado na matriz — devendo como é óbvio ser superior à última oferta.

Quando a subrotina do licitador secreto é acedida, este oferecerá até $50 por um artigo cujo valor atinja os $100. Isto permite aos jogadores com pouco capital aumentarem-no apropriando-se de pequenos lotes — se os outros deixarem.

No caso de artigos com valor superior a $100, o licitador secreto recusar-se-á a fazer qualquer oferta em 10% dos casos, e em 25% dos casos continuará a fazer as suas ofertas depois de o seu valor ter sido atingido. Tendo em conta estas duas possibilidades, é óbvio que interessa aos jogadores aumentar as ofertas de pequenas quantias, dando assim ao licitador secreto mais possibilidades de sair. O valor da oferta secreta é definido nas linhas 1080 e 1100; foi pensado de tal modo que não seja excessivamente grande em relação à última oferta realizada, tendo no entanto em consideração o verdadeiro valor do artigo.

A licitação continua até ficar apenas um jogador, secreto ou não. Os licitadores secretos não podem comprar barato; isto é, no seu caso o preço de aquisição é sempre pelo menos equivalente a 75% do verdadeiro valor do objecto. Isto não se aplica aos jogadores, no entanto. Pelo menos em teoria, é possível a um jogador adquirir um artigo com um valor de $10 000 por apenas $1.

Lista de variáveis

Q$	String de entradas
L	Número de lote actual, 1-20
P	Número de jogadores
X	Contador
P$(X)	Array com nomes dos jogadores
P(X,Y)	Matriz dos jogadores, onde X = jogador e Y = 1 = dinheiro restante 2 = oferta (-9, ausência de oferta)
E	Flag indicadora do final do jogo
C	Idade atribuída ao artigo
V	Valor real do artigo
B	Valor da última oferta
B1	Valor da oferta secreta
B2	Número de jogadores que ainda apostam
B9	Número do jogador com maior oferta
P1	Número de licitação do jogador secreto

Listagem do programa

```
0010 PRINT TAB(20); "LEILÃO";
0020 PRINT TAB(20);"= = = = ="
0030 INPUT "Deseja instruções",Q$
0040 IF LEFT$(Q$,1)<>"S" THEN 290
0050 PRINT "Pode jogar qualquer número de pessoas. Cada ";
0060 PRINT " uma possui inicialmente $100. São leiloados ";
0070 PRINT "vários artigos, podendo valer milhares ou";
0080 PRINT " nada — e é difícil saber o seu verdadeiro ";
0090 PRINT "valor. É pedida uma oferta a cada jogador. ";
```

```
0100 PRINT " Escreva 0 se não quiser oferecer. As ofer
tas ";
0110 PRINT "continuam até ficar apenas um jogador, que ";
0120 PRINT " obviamente fica com o artigo. "
0140 PRINT "Note que existe um licitador 'secreto' que ";
0150 PRINT " conhece o verdadeiro valor do artigo; mas ";
0160 PRINT "tenha cuidado — pode tentar enganá-lo!"
0170 INPUT "Carregue em RETURN...",Q$
0180 PRINT "Quando é vendido um artigo, é dito ao seu ";
0190 PRINT " comprador o valor do que comprou. O dinhei
ro ";
0200 PRINT " em seu poder é modificado, passando-se ao lo
te ";
0210 PRINT " seguinte."
0220 PRINT " O vencedor é aquele que terminar com mais
dinheiro."
0230 PRINT:PRINT :INPUT "Carregue em RE
TURN...",Q$
0240 DATA Jarrão, Quadro, Vaso de flores, Aparador,
Objecto artístico
0250 DATA Escultura, Quadro, Manuscrito, Suporte de
pautas, Violino
0260 DATA Peça de cerâmica, Prato, Desenho, Relógio,
Lanterna
0270 DATA Taça de prata, Cadeira, Mesa pequena, Pistola,
Espada
0290 PRINT :L=1
0300 REM — Início
0310 PRINT "Quantos jogadores";
0320 INPUT P
0330 IF P<1 THEN 310
0340 PRINT "Por favor escreva o nome pelo qual quer ser ";
0350 PRINT " conhecido pelo leiloeiro."
0360 FOR X=1 TO P:PRINT
0370 PRINT "Jogador n.";X;
0380 INPUT P$(X):IF P$(X)="THEN PRINT "Repita":-
GOTO 370
0390 NEXT X
0400 DIM P(P,2):REM — Para cada jogador, dinheiro e
aposta
```



```
0410 FOR X=1 TO P:P(X,1)=100:P(X,2)=-9:NEXT X
0420 REM — Menos 9 indica ausência de aposta
0499 REM — Relatório aos jogadores
0500 PRINT
0520 FOR X=1 TO P
0530 PRINT P$(X);" — Você tem $";P(X,1)
0540 NEXT X:PRINT:PRINT:PRINT:IF E=1 THEN END
0550 PRINT "Boa tarde, senhoras e senhores."
0560 PRINT "Apresento agora o lote n.";L; ". "
0570 RESTORE
0599 REM — Começa o leilão
0610 READ L$:PRINT L$; "atribuído(a) ao século ";
0620 C=INT(RND(0) *6+14)
0630 PRINT C;" Que oferecem por este artigo raro?"
0640 REM — Determinar valor verdadeiro
0650 V=INT(RND*RND*RND*10 000+1)
0660 B=0:REM — Maior oferta
0670 B2=P:REM — Número de jogadores que oferecem
0680 B1=0:REM — Oferta secreta
0690 B9=0:REM — Número do jogador com maior oferta
0698 REM — Loop de oferta
0699 REM — Determinar posição do licitador secreto
0700 P1=(INT(RND(0)*P+1)
0710 FOR X=1 TO P:IF X<>P1 THEN 740
0720 IF B=-9 THEN 740:REM — Não interessado
0730 GOSUB 1000
0740 IF P(X,2)=0 THEN 930
0750 IF X=B9 THEN 930
0760 IF B<>0 THEN 800
0770 PRINT
0780 PRINT P$(X);" — Qual é a sua primeira oferta";
0790 GOTO 810
0800 PRINT :PRINT "Ofereceram $";B;";a sua oferta, "
;P$(X);
0810 INPUT P(X,2)
0820 IF P(X,2)<>0 THEN 860
0830 PRINT "Obrigado, ";P$(X);" — não oferece."
0840 B2=B2-1:GOTO 390
0860 IF P(X,2)<=P(X,1)THEN 880
0870 PRINT "Mas apenas possui $";P (X,1)"!":GOTO 760
0880 IF P(X,2)>B THEN 910
```

```
0890 PRINT “Já tenho uma oferta de $”;B;“ — por favor ”;
0900 PRINT “ ofereça mais, ou não ofereça.”:GOTO 760
0910 B=P(X,2):B9=X:REM — Jogador
0920 PRINT “$”;B;“ — Obrigado, ”;P$(X)
0930 NEXT X:IF B2>1 THEN 700
0940 IF B1=-9 THEN 1200:REM - O último jogador compra
0950 IF B2=0 THEN 1200:REM — Só fica o jogador secreto
0960 IF B<>B1 THEN GOSUB 1000
0970 GOTO 700
0999 REM — Subrotina do jogador secreto
1000 IF B>=V THEN 1050
1010 IF  V<100  THEN  B1=INT(RND(0)*50+1):GOTO
1104
1020 REM — Confundir não oferecendo?
1030 IF RND(0)>.9 THEN B1=-9:RETURN
1040 GOTO 1080
1050 REM — Confundir oferecendo demais?
0160 IF RND(0)>.25 THEN 1130
1070 REM — Calcular oferta
1080 B1=INT(((ABS(V—B)*RND(0)/3.3)+B)/10)*10
1090 IF B1>B THEN 1110
1100 IF  B1<V*1.5  THEN  B1=INT((B1+RND(0)*100+1)-
/10)*10:GOTO 1090
1102 V1=-9:GOTO 1130
1104 IF B1<= B THEN 1130
1110 PRINT :B=B1:B9=9
1120 PRINT “$”;B;“ Uma oferta anónima.”
1130 RETURN
1190 REM — Vender pela maior oferta
1200 IF B9<>9 THEN 1250
1210 IF B1>V*.75 THEN 1250
1220 PRINT :PRINT “O artigo é retirado. Havia ”:
1230 PRINT “ um preço de reserva de $”;V
1240 GOTO 1400
1250 PRINT :PRINT
1260 PRINT “Mais nenhuma oferta? Não? ”
1270 PRINT TAB(12);
1280 IF B9<>9 THEN 1310
1290 PRINT  “Arrematado  pela  oferta  anónima  de  $”;B:
GOTO 1310
```



```
1300 PRINT “Arrematado por ”;P$(B9);“ por $”;B
1310 PRINT :PRINT
1320 PRINT “O valor real do artigo é $”;V
1330 IF V>B THEN 1350
1340 PRINT “É pena, ”;:GOTO 1360
1350 PRINT “Foi uma pechincha, ”;
1360 IF B9=9 THEN PRINT “senhor!”GOTO 1400
1370 PRINT P$(B9);“!”:P(B9,1)=P(B9,1)—B+V
1380 PRINT  “As  suas  disponibilidades  são  agora  de  $”;
P(B9,1)
1390 IF P(B9,1)=0 THEN 9000
1399 REM — Preparar para o lote seguinte
1400 FOR X=1 TO P:P(X,2)=-9:NEXT X
1410 L=L+1:IF INT (L/20)*20=L THEN 8000
1420 PRINT TAB(12); “= = = = = = = = = = =”:PRINT
1430 PRINT “Em seguida apresento o lote n.”;L;“. ”;
1440 GOTO 610
7999 REM — Continuar o leilão?
8000 PRINT :PRINT:PRINT
8010 PRINT “Neste momento, senhoras e senhores, ”;
8020 PRINT “faremos um intervalo para ir ao bar.”
8030 PRINT
8040 INPUT “Desejam continuar”,Q$
8050 IF LEFT$(Q$,1)<>“S” THEN E=1:GOTO 500
8060 PRINT :L=L+1:GOTO 500
8999 REM — Final do jogo
9000 PRINT “Terminou o leilão.”
9010 PRINT “Pontuações finais —”
9020 PRINT :E=1:GOTO 500
```

Exemplo de execução

Deseja instruções? Não

Quantos jogadores?
Por favor escreva o nome pelo qual quer ser
conhecido pelo leiloeiro.
Jogador n.1? Tomás

Jogador n.2? João
Jogador n.3? Manuel

Tomás — você tem $100
João — você tem $100
Manuel — você tem $100

Boa tarde, senhores e senhoras.
Apresento agora o lote n.1. Jarrão atribuído(a) ao século 15. Que oferecem por este artigo raro?

Tomás — Qual é a sua primeira oferta1
$1 — Obrigado, Tomás

$6 — Uma oferta anónima.
$7 — Obrigado, João
Ofereceram $7; a sua oferta, Manuel8
$8 — Obrigado, Manuel

Ofereceram $8; a sua oferta, Tomás10
$10 — Obrigado, Tomás

Ofereceram $10; a sua oferta, João0
Obrigado, João — não oferece.

$25 — Uma oferta anónima.

Ofereceram $25; a sua oferta, Manuel30
$30 — Obrigado, Manuel

Ofereceram $30; a sua oferta, Tomás0
Obrigado, Tomás — não oferece.
$110 — Uma oferta anónima.

Ofereceram $110; a sua oferta, Manuel0
Obrigado, Manuel — não oferece
Mais nenhuma oferta? Não?
Arrematado pela oferta anónima de $110

O valor real do artigo é $26
É pena, senhor!

= = = = = = = = = = = = = = = =

Em seguida apresento o lote n. 2. Quadro atribuído (a) ao século 18. Que oferecem por este artigo raro?

Tomás — Qual é a sua primeira oferta1
$1 — Obrigado, Tomás

$90 — Uma oferta anónima.

Ofereceram $90; a sua oferta, João0
Obrigado, João — não oferece.

Ofereceram $90; a sua oferta, Manuel0
Obrigado, Manuel — não oferece

Ofereceram $90; a sua oferta, Tomás0
Obrigado, Tomás — não oferece.
$210 — Uma oferta anónima.

O artigo é retirado. Havia um preço de reserva de $602

= = = = = = = = = = = = = = = =

Em seguida apresento o lote n.3. Vaso de flores atribuído(a) ao século 16. Que oferecem por este artigo raro?

Tomás — qual é a sua primeira oferta1
$1 — Obrigado, Tomás

$20 — Uma oferta anónima.

Ofereceram $20; a sua oferta, João22
$22 — Obrigado, João

Ofereceram $22; a sua oferta, Manuel0
Obrigado, Manuel — não oferece.

Ofereceram $22; a sua oferta, Tomás0
Obrigado, Tomás — não oferece.

$70 — Uma oferta anónima.

Ofereceram $70; a sua oferta, João75
$75 — Obrigado, João

$80 — Uma oferta anónima

$150 — Uma oferta anónima.

Ofereceram $150; a sua oferta, João0
Obrigado, João — não oferece.

O artigo é retirado. Havia um preço de reserva de $345

= = = = = = = = = = = = = = =

Ofereceram $1; a sua oferta, João2
$2 — Obrigado, joão

$10 — Uma oferta anónima.

Ofereceram $10; a sua oferta, Manuel20
$20 — Obrigado, Manuel

Ofereceram $20; a sua oferta, Tomás0
Obrigado, Tomás — não oferece.

$70 — Uma oferta anónima.

Ofereceram $70; a sua oferta, João77
$77 — Obrigado, João

Ofereceram $77; a sua oferta, Manuel0
Obrigado, Manuel — não oferece.

Mais nenhuma oferta? Não?
Arrematado por João por $77

O valor real do artigo é $130
Foi uma pechincha, João!
As suas disponibilidades são agora de $153

= = = = = = = = = = = = = = =

Em seguida apresento o lote n. 18. Mesa pequena atribuído(a) ao século 17. Que oferecem por este artigo raro?

$30 — Uma oferta anónima
Ofereceram $30; a sua oferta, Tomás33
$33 — Obrigado, Tomás

Ofereceram $33; a sua oferta, João0
Obrigado, João — não oferece.

Ofereceram $33; a sua oferta, Manuel0
Obrigado, Manuel — não oferece.

$70 — Uma oferta anónima.

Ofereceram $70; a sua oferta, Tomás77
$77 — Obrigado, Tomás

$90 — Uma oferta anónima.

$110 — Uma oferta anónima.

Ofereceram $110; a sua oferta, Tomás0
Obrigado, Tomás — não oferece.

O artigo é retirado. Havia um preço de reserva de $205.

= = = = = = = = = = = = = = =

Apresento agora o lote n. 19. Pistola atribuído(a) ao século 18. Que oferecem por este artigo raro?

Tomás — qual é a sua primeira oferta1
$1 — Obrigado, Tomás

Ofereceram $1; a sua oferta, João2
$2 — Obrigado, João

Ofereceram $2; a sua oferta, Manuel3
$3 — Obrigado, Manuel

Ofereceram $3; a sua oferta, Tomás5
$5 — Obrigado, Tomás

Ofereceram $5; a sua oferta, João0
Obrigado, João — não oferece.

Ofereceram $5; a sua oferta, Manuel0
Obrigado, Manuel — não oferece.

Mais nenhuma oferta? Não?

Arrematado por Tomás por $5

O valor real do artigo é $1002
Foi uma pechincha, Tomás!
As suas disponibilidades são agora de $1097

Neste momento, senhoras e senhores, faremos um intervalo para ir ao bar.

Desejam continuar? N

Pontuações finais —

Tomás — você tem $1907
João — você tem $153
Manuel — você tem $100

XXIII

WOTSIT? (2,1 K)

"Wotsit?" é um jogo que qualquer pessoa achará interessante. São omitidos sinais de mais e menos em somas, sendo necessário ao jogador colocá-los de novo. Pode-se especificar quantos operadores se pretende utilizar, dado que são possíveis até 10, e o resultado constituirá certamente um bom teste de agilidade mental. As crianças conseguirão resolver problemas com dois ou três sinais em falta, e o adulto normal suportará facilmente quatro ou cinco.

Se bem que o problema seja simples de enunciar, a escrita do programa não é tão fácil. Existem dois problemas principais na programação. Em primeiro lugar, muitas combinações de valores e operadores darão resultados semelhantes, principalmente quando são repetidos números. Por exemplo, 17−3+3 dá o mesmo resultado que 17+3−3, pelo que não bastará fazer uma simples comparação dos sinais indicados pelo jogador com os conhecidos pela máquina. É necessário reconstruir o total a partir dos números originais e dos sinais indicados pelo jogador; se ambos forem iguais o computador deve aceitar a resposta.

Isto significa por sua vez que é necessário armazenar cada número original para ser usado mais tarde. Isto leva-nos a considerar o segundo problema, que consiste em não podermos construir uma string de números e sinais para processamento imediato, sendo forçados a fazer uma soma contínua, número a número.

Pensei em incluir sinais de multiplicação e divisão, mas depois de pensar um pouco convenci-me de que o programa resultante seria enormemente complicado devido a problemas de precedência das operações (não esqueça que o computador realiza as multiplicações e divisões antes das somas e subtracções). Se quiser

fazer esse programa, irá enfrentar bastantes dificuldades. Aliás o programa já é bastante difícil para qualquer jogador nesta versão.

Lista de variáveis

Q$	String de entradas
R	Problemas respondidos correctamente
S	Número total de somas
M	Número de operadores em falta
P	Número de problema, 1 a 3
P$	String de operadores em falta
D(X)	Números usados
T	Total do processamento original
N	Contagem de números
O	Número de operador omitido, 1 a 2 onde 1 = somar 2 = subtrair
I	Contagem dos operadores indicados pelo jogador
A$	Operador indicado pelo jogador
T1	Total da soma a partir dos operadores do jogador
O$	String de todos os operadores do jogador.

Listagem do programa

```
0010 PRINT "WOTSIT?"
0020 PRINT "= = = = = ="
0030 PRINT :PRINT
0040 INPUT "Deseja instruções",Q$
0050 IF LEFT$(Q$,1)="S" GOSUB 1000
0060 PRINT :R=0:S=0
0070 INPUT "Quantos operadores em falta".Q$
0080 M=ABS(INT(M))
0090 IF M>10THEN 70
0100 IF M=0 THEN 70
0110 REM — Dar três elementos ao jogador
0120 FOR P=1 TO 3:P$="":S=S+1
0130 REM Dar primeiro número
```



```
0140 D(1)=INT(RND(O)*20+1):T=D(1)
0150 PRINT D(1);" ? ";
0160 REM — Dar números iguais aos operadores
0170 FOR N=2 TO M+1
0180 D(N)=INT(RND(O)*20+1):PRINT D(N);
0190 REM Escolher operador
0200 O=INT(RND(O)*2+1)
0210 IF O=1 THEN P$=P$+" + ":T=T+D(N)
0220 IF O=2 THEN P$=P$+"–";T=T–D(N)
0230 IF N<>M+1 PRINT " ? ";
0240 NEXT N:PRINT " = ";T
0250 PRINT:PRINT "Quais são os operadores em falta?"
0255 PRINT "Carregue em RETURN depois de cada
caracter"
0260 O$="":FOR I=1 TO M
0270 INPUT A$
0280 IF A$=" + " THEN 310
0290 IF A$="–" THEN 310
0300 PRINT "Indique erro. Por favor escreva '–' ou '+'":
GOTO 270
0310 O$=O$+A$:NEXT I
0320 IF O$=P$ PRINT "Correcto!":R=R+1:GOTO 400
0325 REM — Reconstruir a partir da entrada
0330 T1=D(1):FOR X=2 TO M+1
0340 IF MID$(O$,X–1,1)=" + " THEN T1=T1+D(X)
0350 IF MID$(O$,x–1,1)="–"THEN T1=T1–D(X)
0360 NEXT X:IF T=T1 PRINT "Correcto!":R=R+1:
GOTO 400
0370 PRINT "Errado. A resposta era: ";
0380 FOR X=1 TO N:PRINT MID$(P$,X,1):" ";:NEXT X
0390 PRINT
0400 NEXT P:PRINT:PRINT
0410 PRINT "A sua pontuação até agora é";R;"num total
de ";S
0420 PRINT "Isto é, ";R/S*100; "Por cento."
0430 INPUT "Novo jogo (S-N)",Q$
0440 IF Q$="S" THEN PRINT:GOTO 70
0450 END
1000 PRINT "Neste jogo vou apresentar-lhe séries de somas ";
1010 PRINT " e subtracções, sem os respectivos sinais."
```

```
1020 PRINT "Por exemplo -"
1030 PRINT " 3 ? 4 ? 1 = 6"
1040 PRINT "Deverá escrever os sinais em falta — neste ";
1050 PRINT " caso, + e -, porque 3 + 4 - 1 = 6."
1060 PRINT "Vou perguntar-lhe quantos operadores deseja,
";
1070 PRINT "  ou seja, quantos sinais em falta. Não seja  ";
1080 PRINT "ambicioso de início... Boa sorte!"
1090 INPUT "Por favor carregue em RETURN...",Q$
1100 RETURN
```

Exemplo de execução

```
Deseja instruções? NÃO.

Quantos operadores em falta? 2
14 ? 9 ? 2 = 25

Quais são os operadores em falta?
Carregue em RETURN depois de cada caracter
+
+
Correcto!
17 ? 12 ? 13 = -8

Quais são os operadores em falta?
Carregue em RETURN depois de cada caracter
-
-
Correcto!
16 ? 4 ? 7 = 13

Quais são os operadores em falta?
Carregue em RETURN depois de cada caracter
+
-
Correcto!
```



```
A sua pontuação até agora é 3 num total de 3
isto é, 100 por cento.
Novo jogo (S-N)? S

Quantos operadores em falta? 3
4 ? 10 ? 10 ? 16 = 0

Quais são os operadores em falta?
Carregue em RETURN depois de cada caracter
-
-
+
Correcto!
3 ? 18 ? 12 ? 16 = 25

Quais são os operadores em falta?
Carregue em RETURN depois de cada caracter
-
-
+
Correcto!
9 ? 6 ? 11 ? 8 = 34

Quais são os operadores em falta?
Carregue em RETURN depois de cada caracter
-
+
-
Correcto!

A sua pontuação até agora é 6 num total de 6
isto é, 100 por cento
```

XXIV

CASA ASSOMBRADA (10 K)

Apresentamos em seguida um jogo que certamente agradará aos membros mais jovens da família! Podem explorar uma casa assombrada, abrir caixas e aparadores enquanto o fazem. Na maior parte deles encontrarão coisas sem interesse, mas em alguns estarão objectos valiosos. Ssshhhh — não diga nada a ninguém — num está uma caveira que salta para fora e nos apanha!

Algumas caixas e aparadores encontram-se fechados, sendo necessária uma ferramenta qualquer para os abrir; mas é possível encontrar ferramentas e chaves noutras caixas ou gavetas. A certo momento são até necessárias duas chaves ou ferramentas — é nesse local que se encontra o tesouro mais valioso! O leitor terá de correr toda a casa, com o coração na boca, abrindo gavetas a medo, esperando encontrar as chaves e o tesouro mas sempre temendo encontrar a caveira em vez dele. É óptimo!

O jogo termina quando a caveira nos apanha, ou quando nos afastamos da casa. Podemos afastar-nos da casa caminhando para norte a partir da parede norte, para oeste a partir da parede oeste, etc. Quando tal acontece, é dito ao jogador o valor daquilo que traz consigo, e a sua pontuação. Consegue mais pontos quando atinge o tesouro em menos movimentos, e nada pontua se não encontrar nenhum objecto valioso.

Se o computador dispuser de uma memória suficiente, as descrições podem ser ampliadas (linhas 1080-1280), podendo-se também incluir um "ruído" ou um qualquer "gerador" de acontecimentos que ajudem a manter o ambiente; por exemplo portas batendo, ruído de correntes, gargalhadas, etc. Melhor ainda, se dispuser dos meios necessários, pode gravar estes ruídos e reproduzi-los num altifalante!

Se no entanto a memória de que a máquina dispõe é limitada, retire uma ou duas salas e as respectivas descrições, além do número correspondente de objectos inúteis. Existem 20 de cada, pelo que mesmo no caso de apenas remover a cave e o espaço no topo das escadas que conduzem a esta é possível poupar 1 K.

O programa propriamente dito começa na linha 200, onde são distribuídos os primeiros 11 objectos aleatoriamente pelas 20 salas ou locais. Conseguimos isto utilizando uma matriz B, na qual para cada sala temos três parâmetros. O primeiro é o número do objecto aí escondido. Os últimos nove objectos só podem ser acedidos usando uma chave, pelo que depois de colocarmos o número na primeira coluna da tabela damos à segunda coluna um número entre 1 e 9, representando uma das 9 chaves. Só no caso do objecto 20 (o tesouro) se utiliza uma segunda chave, indicada na coluna 3.

O array A$ recebe o nome de cada artigo, lido das instruções DATA nas linhas 160 a 190. A segunda matriz M define os movimentos, sendo novamente concebida como uma tabela. Para cada sala por onde se passa, as seis colunas da tabela dão os números das salas para onde podemos passar se nos deslocarmos para norte, sul, leste, oeste, para cima ou para baixo, respectivamente. Estas instruções DATA são lidas das posições adjacentes às descrições da sala, o que é o local mais lógico para as colocar e permite facilitar a escrita e a compreensão do programa. Note que aqui se ilustra claramente a possibilidade de espalhar as instruções DATA pela listagem.

Só existem 20 salas ou locais e 21 valores possíveis da matriz M. O "local" 21 equivale a sair da casa, terminando o jogo. O "local" 22 não tem saída, e o utilizador recebe a mensagem apropriada se tentar esse movimento.

Chamo ainda a atenção do leitor para o método usado para descodificar a entrada do utilizador. Este pode usar 7 ordens: as seis direcções e ABRIR. A subrotina que se inicia na linha 2000 permite dar entrada a toda a palavra ou apenas à inicial — o que requer uma cuidadosa escolha das ordens possíveis! Se a ordem é legal, a subrotina atribui um valor 1 a 7 à variável X, correspondendo o número à ordem dada. Este método de descodificação das ordens é bastante melhor do que os utilizados até aqui. Como é óbvio, se o quisermos, poderemos aceder a esta subrotina em qualquer ponto do programa.

Finalmente, o array A(X) contém o número de objectos recolhidos — o inventário — sendo usado para verificar se podemos abrir uma gaveta fechada. A linha 590 soma o valor dos objectos valiosos a um total, T.

Lista de variáveis

B(X,Y)	Matriz de objectos, onde X = localização Y = 1 número do objecto 2 = Objecto para abrir, se houver 3 = Segundo objecto para abrir, se houver
A$(X)	Array de nomes dos objectos
M(X,Y)	Matriz de movimentos, onde X = posição actual Y = 1 = sala a norte 2 = sala a sul 3 = sala a leste 4 = sala a oeste 5 = sala por cima 6 = sala por baixo
N	Número da sala ou local actual
A$	String de respostas
X	Contador; também número de movimento
A	Número dos objectos
R	Número das salas
Y	Contador geral
K	Número de objecto para abrir
F	Flag; 0 = receptáculo fechado; 1 = aberto
N9	Número de objectos recolhidos
N1 e N2	Factores temporários de N
R$	String de iniciais permissíveis

Listagem do programa

```
0010 PRINT "CASA ASSÒMBRADA"
0020 PRINT "= = = = = = = = = = = ="
0030 DIM B(20,3),A$(20),M(20,6):N = 1
```



```
0040 LINE = 72 :INPUT "Deseja instruções",A$
0050 IF LEFT$(A$,1) = "N" THEN 140
0060 PRINT "Vai explorar uma casa assombrada, encontrando ";
0070 PRINT " talvez objectos valiosos dentro dela. Em cada ";
0080 PRINT " movimento deve escrever uma das seguintes ordens: ";
0090 PRINT "N (norte), S (sul), E (este) ou O (oeste). Pode ";
0100 PRINT " também escrever C (para cima) ou B (para baixo). " ;
0110 PRINT "Se quiser abrir qualquer coisa, escreva A."
0130 PRINT "Por favor carregue em RETURN para começar. ";A$
0135 REM — Dar algo para ler
0140 GOSUB 1000
0150 FOR X = 1 TO 20:READ A$(X):NEXT X
0160 DATA chave de parafusos, martelo, chave pequena, chave de latão, chave grande
0170 DATA lata de óleo, serra, fita de chapéu, lima, lupa, rato morto
0180 DATA pão duro, fotografia, jornal velho, caveira, saco de moedas
0190 DATA conjunto de colheres, maço de notas, anel de rubi, tesouro
0200 PRINT :PRINT "Um momento por favor..."
0210 REM — distribuir objectos
0220 REM — Um objecto por sala
0230 For A = 1 TO 11
0240 R = INT(RND(0)*20 + 1)
0250 IF B(R,1)<>0 THEN 240
0260 B(R,1) = A;NEXT A
0270 REM — Nos últimos 9, aplicar uma chave
0280 FOR A = 12 TO 19
0290 R = INT(RND(0)*20 + 1)
0300 IF B(R,1)<>0 THEN 290
0310 B(R,1) = A:B(R,2) = INT(RND(0)*9 + 1)
0320 NEXT A
0330 REM — Para o tesouro, duas chaves
0340 FOR R = 1 TO 20 :IF B(R,1) = 0 THEN 350:NEXT R
0350 B(R,1) = 20:B(R,2) = INT(RND(0)*9 + 1)
```

```
0360 B(R,3)=INT(RND(0)*9+1):IF B(R,3)=B(R,2) THEN
360
0370 REM — Leitura dos dados
0380 FOR X=1 TO 20:FOR Y=1 TO 6:READ M(X,Y):
NEXT Y:NEXT X
0390 REM — Movimento
0400 PRINT :GOSUB 2000:M=M+1
0410 IF X<>7 THEN 610
0420 REM — Ordem de abrir
0430 IF B(N,1)=0 GOSUB 1300:GOTO 400
0440 REM — Necessária chave?
0450 IF B(N,2)=0 THEN 520
0460 K=B(N,2):GOSUB 4000:REM — Verificar inventário
0470 IF F=0 THEN 400
0480 REM Segunda chave?
0490 IF B(N,3)=0 THEN 520
0500 K=B(N,3):GOSUB 4000:REM — Verificar inventário
0510 IF F=0 THEN 400
0520 X=B(N,1):IF X<>15 THEN 550
0530 PRINT "AAAAAGH! Uma caveira! E mexe-se!"
0540 PRINT "Pouca sorte...":END
0550 PRINT "Está um(a) ";A$(X);" aqui. Quer este
objecto?"
0560 INPUT Q$:IF LEFT$(Q$,1)="N" THEN 400
0570 B(N,1)=0
0580 N9=N9+1:A(N9)=X:PRINT "O.K. — é seu."
0590 IF X>=16 THEN T=T+10*X
0600 GOTO 400
0605 REM — Verificar movimento
0610 IF M(N,X) =21 THEN N=M(N,X):GOSUB 1000:
GOTO 400
0620 GOSUB 1290:GOTO 400
1000 IF M=0 GOSUB 3000
1005 N1=INT(N/5):N2=N-N1*5
1010 ON N1+1 GOTO 1020,1030,1040,1050,1060
1020 ON N2 GOTO 1080,1090,1090,1100,1110
1030 ON N2+1 GOTO 1120,1130,1140,1150,1160
1040 ON N2+1 GOTO 1170,1180,1190,1200,1210
1050 ON N2+1 GOTO 1220,1230,1240,1250,1260
1060 ON N2+1 GOTO 1270,1280
```



```
1080 DATA 22,21,2,4,22,22
1081 PRINT "Você está junto ao lado sul de uma mansão do ";
1082 PRINT " século passado, virado para a porta. No canto ";
1083 PRINT "direito vê-se uma torre. A seu lado está um ";
1084 PRINT " túmulo em pedra. Talvez possa abri-lo. ":
RETURN
1090 DATA 3,1,21,10,22,22
1091 PRINT "Você está junto ao lado leste da casa, com a ";
1092 PRINT " torre à esquerda. Vê uma janela partida
por ";
1093 PRINT "onde poderia entrar. Abaixo da janela está ";
1094 PRINT " uma velha caixa de ferramentas de
jardinagem.": RETURN
1100 DATA 21,12,2,4,22,22
1101 PRINT "Atrás da casa, no lado norte, você vê uma ";
1102 PRINT "ala do alpendre. Ao lado da porta traseira ";
1103 PRINT " vê um caixão.": RETURN
1110 DATA 3,1,15,21,22,22
1111 PRINT "Na parede oeste da casa está uma janela aberta."
1112 PRINT "Uma trepadeira cobre em parte um interessante ";
1113 PRINT " paiol de carvão. A trepadeira permite-lhe
entrar ";
1114 PRINT " facilmente pela janela. Escreva E para o
fazer ";
1115 PRINT " ou A para abrir o paiol.": RETURN
1120 DATA 16,22,6,22,22,22
1121 PRINT "Está na sala de desenho. A poeira cobre as ";
1122 PRINT " cadeiras, a mesa, e a caixa de aspecto ca
ro. ";
1123 PRINT "Existem duas portas, a norte e leste."
1124 RETURN
1130 DATA 13,22,7,5,19,22
1131 PRINT " Está no hall, rodeado por paredes almofada
das. Uma ";
1132 PRINT "escada sobe na escuridão. Há portas para
norte, ";
1133 PRINT " sul, leste e oeste, mas a de sul é a porta ";
1134 PRINT "principal. Note que existe um armário debaixo
";
1135 PRINT " das escadas. "
```

```
1136 RETURN
1140 DATA 9,22,8,6,22,22
1141 PRINT "Uma sala de estar, sem mobília excepto um ";
1142 PRINT " armário num canto. Há portas para norte, ";
1143 PRINT "oeste e leste.": RETURN
1150 DATA 22,22,22,7,22,22
1151 PRINT "Uma sala de música, com um piano velho e um ";
1152 PRINT " suporte de pautas. O espaço circular num dos ";
1153 PRINT " cantos, com uma janela grande, forma a base ";
1154 PRINT " da torre. Uma porta para oeste."
1155 RETURN
1160 DATA 11,7,22,13,22,22
1161 PRINT "Sala de jantar, com janelas a norte e a leste. ";
1162 PRINT " A virada a norte está semi-aberta. Numa
parede ";
1163 PRINT " está uma mesa com um bolo bolorento por ci
ma.";
1164 PRINT "Ao lado encontra-se um armário."
1165 PRINT "Existem duas portas, uma para sul e outra
para ";
1160 PRINT "oeste.":RETURN
1170 DATA 11,22,2,22,22,22
1171 PRINT "Está na ala leste do alpendre fechado. Existe ";
1172 PRINT " uma janela partida para o exterior, e uma ";
1173 PRINT "grande caixa de estanho.":RETURN
1174 PRINT "a extremidade norte.":RETURN
1180 DATA 22,9,10,12,22,22
1181 PRINT "Na ala norte do alpendre. Existe uma porta ";
1182 PRINT " para oeste e uma janela aberta para sul. ";
1183 PRINT "Vê-se uma caixa de brinquedos muito grande."
1184 RETURN
1190 DATA 3,13,11,22,22,22
1191 PRINT "Está na porta traseira, aberta a norte. ";
1192 PRINT " Para sul entra na casa, e existe ";
1193 PRINT "outra porta para leste. Vê uma velha arca. "
1194 RETURN
1200 DATA 12,6,9,14,22,22
1201 PRINT "Está num corredor escuro com uma porta em
cada ";
```



```
1202 PRINT " lado. No chão vê uma caixa de estanho."
1203 RETURN
1210 DATA 22,17,13,15,22,22
1211 PRINT "Está na cozinha, com algumas manchas
estranhas ";
1212 PRINT " no chão. Existem portas para leste, sul e ";
1213 PRINT "oeste e um grande fogão em ferro.":RETURN
1220 DATA 22,22,14,4,22,22
1221 PRINT " Uma despensa apertada, com prateleiras dos ";
1222 PRINT " dois lados. Existe uma porta para leste e uma ";
1223 PRINT " janela partida a oeste. Vê um velho armá
rio ";
1224 PRINT "enferrujado.":RETURN
1230 DATA 22,5,22,22,22,22
1231 PRINT "Um estúdio com cortinas pesadas e cheio de ";
1232 PRINT " poeira. Uma porta para sul. No centro, uma ";
1233 PRINT " secretária grande e pesada, com gavetas."
1234 RETURN
1240 DATA 14,22,22,22,22,18
1241 PRINT "Um pequeno espaço no topo das escadas para a ";
1242 PRINT " cave. Uma porta a norte, com um peque
no ";
1243 PRINT "armário a norte.":RETURN
1250 DATA 22,22,22,22,17,22
1251 PRINT "Está agora na cave. Apenas vê um baú velho."
1252 RETURN
1260 DATA 22,22,20,22,22,6
1261 PRINT "Está na escuridão, no topo das escadas. Uma ";
1262 PRINT "porta para leste. Existe uma pequena arca."
1263 RETURN
1270 DATA 22,22,22,19,22,22
1271 PRINT "Está no quarto de dormir da torre. A cama ";
1272 PRINT "desfez-se à muito num monte de poeira, mas a
";
1273 PRINT "cómoda ainda se mantém.":RETURN
1280 IF T>0 THEN 2080
1285 PRINT "Desiste tão cedo? O.K. — jogaremos noutra
ocasião."
1286 END
1290 PRINT "Não pode ir por aí.":RETURN
```

```
1300 PRINT "Está vazio.":RETURN
2000 INPUT "O seu movimento",Q$
2010 Q$=LEFT$(Q$,1)
2030 P$="NSEOCBA"
2040 FOR X=1 TO 7:IF Q$=MID$(R$,X,1) THEN 2070
2050 NEXT X:PRINT "Não percebo a sua instrução."
2060 GOTO 2000
2070 RETURN
2080 PRINT "O valor das mercadorias que trouxe consigo ";
2090 PRINT " é ";T;" $00."
2100 PRINT "A sua pontuação é ";INT(T*11/M)
2110 PRINT :PRINT:PRINT "Adeus":END
3000 PRINT "Um dia em que estava a passear pela parte ";
3010 PRINT "velha da cidade encontrou um terreno coberto ";
3020 PRINT " de erva e rodeado por um muro alto. Este ";
3030 PRINT "ruíra em dado ponto, pelo que conseguiu ";
3040 PRINT " entrar e, depois de abrir caminho entre a ";
3050 PRINT "vegetação, atingir um caminho de cascalho
que ";
3060 PRINT " o conduziu a uma casa antiga."
3070 RETURN
3990 REM Inventário
4000 FOI I=1 TO N9:IF A(I)=K THEN F=1:RETURN
4010 NEXT I:F=0
4020 PRINT "Necessitará de uma ";A$(K); " para abrir."
4030 F=0:RETURN
```


Exemplo de execução

Deseja instruções? NÃO.
Um dia em que estava a passear pela parte velha da cidade encontrou um terreno coberto de erva e rodeado por um muro alto. Este ruíra em dado ponto, pelo que conseguiu entrar e, depois de abrir caminho entre a vegetação, atingir um caminho de cascalho que o conduziu a uma casa antiga.
Você está junto ao lado sul de uma mansão do século passado, virado para a porta. No canto direito vê-se uma torre. A seu lado está um túmulo em pedra. Talvez possa abri-lo..."

Um momento por favor...

O seu movimento? N
Não pode ir por aí.

O seu movimento? E
Você está junto ao lado leste da casa, com a torre à sua esquerda. Vê uma janela partida por onde poderia entrar.
Abaixo da janela está uma velha caixa de ferramentas de jardinagem.

O seu movimento? A
Necessitará de uma chave pequena para abrir.

O seu movimento? S
Você está junto ao lado sul de uma mansão do século passado, virado para a porta. No canto direito vê-se uma torre. A seu lado está um túmulo em pedra. Talvez possa abri-lo...

O seu movimento? A

Necessitará de uma chave de latão para abrir .

O seu movimento? E
Você está junto ao lado leste da casa, com a torre à esquerda. Vê uma janela partida por onde poderia entrar.
Abaixo da janela está uma velha caixa de ferramentas de jardinagem.

O seu movimento? A
Necessitará de uma chave pequena para abrir.

O seu movimento? N
Atrás da casa, no lado norte, você vê uma ala do alpendre.
Ao lado da porta traseira vê um caixão.

O seu movimento? A
Está um(a) chave grande lá dentro. Quer este objecto?

S
O.K. — é seu.

O seu movimento? S
Está na porta traseira, aberta a norte. Para sul entra na casa, e existe outra porta para leste. Vê uma velha arca.

O seu movimento? A
Está um(a) lima lá dentro. Quer este objecto?
S
O.K. — é seu.

O seu movimento? S
Está num corredor escuro com uma porta em cada lado. No chão vê uma caixa de estanho.

O seu movimento? A
Está um(a) chave de latão lá dentro. Quer este objecto?
S
O.K. — é seu.

O seu movimento? S
Está no hall, rodeado por paredes almofadadas. Uma escada sobe na escuridão. Há portas para norte, sul, leste e oeste, mas a de sul é a porta principal. Note que existe um armário debaixo das escadas.

O seu movimento? A

Está um(a) chave de parafusos lá dentro. Quer este objecto?
S
O.K. — é seu.

O seu movimento? E
Uma sala de estar, sem mobília excepto um armário num canto. Há portas para norte, oeste e leste.

O seu movimento? A
Necessitará de uma lata de óleo para abrir.

O seu movimento? N
Sala de jantar, com janelas a norte e a leste. A virada a norte está semi-aberta. Numa parede está uma mesa com um bolo bolorento por cima. Ao lado encontra-se um armário.
Existem duas portas, uma para sul e outra para oeste.

O seu movimento? A
Está um(a) lupa lá dentro. Quer este objecto?
S
O.K. — é seu.

O seu movimento? O
Está num corredor escuro com uma porta em cada lado.
No chão vê uma caixa de estanho.
O seu movimento? A
Está vazio.

O seu movimento? O
Está na cozinha, com algumas manchas estranhas no chão.
Existem portas para leste, sul e oeste e um grande fogão em ferro.

O seu movimento? A
Está um(a) saco de moedas lá dentro. Quer este objecto?
S
O.K. — é seu.

O seu movimento? O
Uma despensa apertada, com prateleiras dos dois lados.

Existe uma porta para leste e uma janela partida para oeste.
Vê um velho armário enferrujado.

O seu movimento? A
Necessitará de uma serra para abrir.

O seu movimento? O
Na parede oeste da casa está uma janela aberta.
Uma trepadeira cobre em parte um interessante paiol de carvão. A trepadeira permite-lhe entrar facilmente pela janela. Escreva E para o fazer ou A para abrir o paiol.

O seu movimento? A
Está um(a) jornal velho lá dentro. Quer este objecto?
N

O seu movimento? S
Você está junto ao lado sul de uma mansão do século passado, virado para a porta. No canto direito vê-se uma torre. A seu lado está um túmulo em pedra. Talvez possa abri-lo...
O seu movimento? A
Está um(a) tesouro lá dentro. Quer este objecto?
S
O.K. — é seu.

O seu movimento? S
O valor das mercadorias que trouxe consigo é 360$00.
A sua pontuação é 132.

Adeus.

XXV

PISTOLEIRO (10,5 K)

"Pistoleiro" permite a todos aqueles que sonham com o oeste americano viverem (ou morrerem...) algumas das suas fantasias. Podem jogar até seis pessoas, e à medida que morrem podem recomeçar ou ser substituídas.

Quando os jogadores escrevem os seus nomes, são-lhes indicadas as suas características pessoais, em termos de coragem, força, velocidade e pontaria com uma pistola, assim como a sua reputação — que será obviamente nula de início. Os jogadores devem tomar nota destas suas características, que o ajudarão a decidir o melhor modo de enfrentar um inimigo. Por exemplo, tal como seria errado a um fraco desafiar alguém para uma rixa, também aquele que possua uma boa pontaria mas seja lento a disparar deve começar a fazê-lo quando se encontra ainda a grande distância da vítima.

Depois de seis jogadores terem indicado os seus nomes, ou quando se carregou em RETURN depois de ser pedido um nome de um jogador que não existe, cada jogador é alternadamente obrigado a enfrentar um fora-da-lei. Primeiro é definido o local, e em seguida o insulto. Ambos estes aspectos são importantes, porque o local define a distância entre os adversários (além de outras coisas) e o insulto determina a reputação que pode ser ganha. A distância é indicada ao jogador, sendo-lhe ainda perguntado se deseja lutar ou fugir. Neste ponto são definidas as características do fora-da-lei, mas o jogador não pode conhecê-las se bem que a máquina forneça uma pista ao imprimir a reputação do fora-da-lei baseada naquelas.

Se o jogador escolhe fugir porque a situação não lhe é favorável ou porque está ocupado a abrir garrafas ou qualquer outra coisa, são deduzidos pontos da sua coragem. Se este factor diminui para zero é retirado do jogo.

Numa luta, o jogador pode escolher entre várias acções. Se se move para a frente, a distância é reduzida e se esta atinge zero há uma luta corpo a corpo. Se não, a iniciativa passa para o fora-da-lei. Se o jogador escolhe disparar, o computador verifica primeiro se possui balas no carregador. Todos iniciam as lutas com seis balas no carregador, e o computador indica quando é necessário recarregar as armas. Esta recarga ocupa dois lances tanto no caso do jogador como do fora-da-lei.

Em seguida, o computador verifica qual é o pistoleiro mais rápido — o jogador ou o adversário. O mais rápido faz dois disparos. A possibilidade de atingir o adversário é determinada pela distância e a pontaria de cada um dos opositores, em termos de percentagens, e quando um deles é atingido o computador determina aleatoriamente em que parte do corpo. A possibilidade de um tiro na cabeça, causando morte imediata, é de um em nove. Os outros disparos provocam pontos de "danos" em função da gravidade da ferida. Estes pontos são deduzidos da força do jogador ou do fora-da-lei, e se esta chega a zero a vítima morre.

O mesmo fora-da-lei nunca pode aparecer em dois lances consecutivos, mas pode voltar a aparecer mais tarde. Por outro lado, as características dos fora-da-lei não se mantêm; são diferentes de cada vez que aparecem. O leitor pode no entanto, se assim quiser, criar uma nova matriz que dê a cada fora-da-lei características fixas.

Pode-se disparar com um revólver de calibre .45 ou uma espingarda de repetição Remington — o que é decidido pelo jogador em cada caso. Em qualquer dos casos apenas se dispõe de seis balas no carregador. A diferença entre as duas armas consiste em que o revólver faz 1,5 vezes mais danos do que a espingarda, mas está limitado a uma distância eficaz de 100 jardas (cerca de 90 metros). A espingarda tem um rigor de disparo superior de 10%.

As lutas corpo a corpo são um pouco mais simples. A distância deve evidentemente ser nula, e o jogador pode ainda usar o seu revólver se o quiser (podendo ainda falhar, se não tiver pontaria!). O computador escolhe aleatoriamente os golpes usados e, se tiverem êxito, os danos causados. Numa luta deste tipo, no entanto, os jogadores não são mortos quando a sua força se reduz para zero — são simplesmente postos "knock-out", podendo lutar novamente mais tarde. Todos os sobreviventes voltam à sua força total depois de uma cena de tiroteio ou de luta corpo-a-corpo.

O jogador que vence recebe pontos de reputação em função do insulto recebido e do local, por exemplo mais pontos por um insulto de morte num local mais difícil. Recebe também um pequeno aumento de bravura, pelo que a sua confiança aumenta... Quando a bravura chega a 90% ou mais, todas as suas outras características — velocidade, pontaria e força são aumentadas até 20% do que falta para 100%. No entanto, o jogador nunca atinge 100% porque isso significaria ser invencível.

Quando todos os jogadores acabam de enfrentar um fora-da-lei, o computador escolhe dois deles aleatoriamente e coloca-os frente a frente! Aplicam-se as mesmas regras, mas o modo como cada um deles joga depende de si mesmo. Apenas posso dizer que é raro que dois jogadores aceitem uma rixa em que ambos sobrevivem; normalmente um deles não resiste e usa o seu revólver...

Neste ponto, os jogadores que foram "mortos" podem reentrar em jogo, ou ser substituídos por outros. Em nenhum momento podem jogar mais de seis "pistoleiros".

Talvez pareça ao leitor que o jogo tende a ter uma duração indeterminada; no entanto, o jogo termina automaticamente quando apenas resta um jogador ou se um dos que estão em jogo chega aos 500 pontos de reputação. A partir desse momento o jogo torna-se demasiado previsível, dado que as características do jogador estão acima de 90%.

Lista de variáveis

Q$	String de entradas
A(X,Y)	Matriz de combate, onde
	X = Número do combatente
	Y = 1 — Bravura ou coragem
	2 — Velocidade
	3 — Pontaria
	4 — Força
	5 — Balas restantes (999 ao recarregar)
	6 — Número

B(X,Y)	Matriz do jogador, onde
	X = Número do jogador
	Y = 1 — Bravura
	2 — Velocidade
	3 — Pontaria
	4 — Força
	5 — Reputação
N$(X)	Array de nomes, onde
	X = 1 a 6 — jogadores
	7 a 16 — foras-da-lei
X	Contador geral
P	Jogador actual
P1	Número de jogadores sobreviventes
R	Distância; também memória temporária de 0
O	Fora-da-lei actual
L	Número do local; também pontos perdidos
I	Número de insulto
Z	Contador geral
F	Flag de combate; 0 = jogador sobrevive;
	1 = jogador morto
A	Contador geral; também flag de movimentos, onde
	0 — Mover ou recarregar a arma
	1 — Disparar
	2 — Luta corpo-a-corpo
S	Vantagem de número de tiros para o mais rápido
U	Indivíduo que dispara
T	Indivíduo sobre quem se dispara
W$	Nome da arma usada
H	Possibilidade de atingir (percentagem)
W	Número da ferida
M	Multiplicador da ferida
V	Vencedor, se for o jogador

Listagem do programa

```
0010 PRINT :PRINT
0020 PRINT TAB(20);“PISTOLEIRO”
0030 PRINT TAB(20);“= = = = = = =”
0040 PRINT :INPUT “Deseja instruções (S-N)”,Q$
```



```
0050 IF Q$ = “S” GOSUB 9000
0060 PRINT :DIM A(2,6),B(6,5),N$(16)
0070 DATA Pedro Gonzalez, Big Jim Jackson, Snake-Eye
Sam
0080 DATA Mexican Pete, Ike Clancy, Bob Dalton, Jake the
Halfbreed
0090 DATA Holy Joe Daniels, Judge Roy Bean, Mad Mike
McCafferty
0100 REM — Array de jogadores
0110 FOR X = 1 TO 6:PRINT
0120 PRINT “Jogador n.”;X;“ — ”;:GOSUB 5000
0130 IF N$(X)<>“” THEN NEXT X:X = 7
0140 P1 = X-1:REM — Número de jogadores
0150 REM — Acrescentar foras-da-lei ao array
0160 FOR X = 7 TO 16:READ N$(X):NEXT X
0199 REM — Primeiro loop ********
0200 P = 1:REM — Primeiro jogador
0210 IF B(P,1) = 0 THEN 680:REM — Fora de jogo
0220 REM — Não repetir o mesmo fora-da-lei
0230 R = O:GOSUB 5100
0240 O = INT(RND(0)*10 + 7):IF R = O THEN 240
0250 PRINT N$(P);“ — encontrou ”;N$(0);“ ”;
0260 L = INT (RND(0)*6 + 1):ON L GOTO 270,280,290,
300,310,320
0270 PRINT “no Last Chance Saloon.”:GOTO 330
0280 PRINT “na estação de caminho de ferro.”:GOTO 330
0290 PRINT “na rua principal de Dodge City.”:GOTO 330
0300 PRINT “num desfiladeiro.”:GOTO 330
0310 PRINT “fora da cidade.”:GOTO 330
0320 PRINT “nas montanhas.”
0330 PRINT “Ele ”;:I = INT(RND(0)*6 + 1)
0340 ON I GOTO 350,360,370,380,390,400
0350 PRINT “diz que você o insultou.”:
GOTO 410
0360 PRINT “diz que você é um cobarde.”:GOTO 410
0370 PRINT “diz que é mais rápido do que você.”:GOTO
410
0380 PRINT “diz que você lhe roubou o cavalo.”:GOTO 410
0390 PRINT “exige que lhe dê o seu dinheiro.”:GOTO 410
0400 PRINT “diz que você matou o irmão dele.”
```

```
0410 REM — Distância depende essencialmente do local
0420 R = INT(RND(0)*L*L*10 + 3*I)
0430 REM — Qualidades de luta do fora-da-lei
0440 FOR  Z = 1  TO  4:A(2*,Z) = INT(RND(0) × 100 + 1):-
NEXT Z
0450 REM — Reacção do jogador
0460 PRINT "Está à distância de ";R;"jardas."
0470 INPUT "  Foge ou luta (F–L)",Q$
0480 IF Q$ = "L" THEN 600
0490 IF Q$ = "F" THEN 510
0500 PRINT "Por favor, F ou L":GOTO 470
0510 PRINT "Sobrevive para outra ocasião, mas este
pequeno fiasco ";
0520 L = INT(RND(0)*L*L + I*I):B(P,1) = B(P,1)–L
0530 PRINT "  custou-lhe ";L;" pontos de bravura. O seu
valor é agora ";
0540 PRINT B(P,1):IF B (P,1)>0 THEN 680
0550 PRINT "Isto faz de si um tal cobarde que nem merece
";
0560 PRINT "  uma cuspidela. Vou tirá-lo deste jogo."
:B(P,1) = 0
0570 P1 = P1–1:IFP1 = 0 THEN END
0590 GOTO 680
0599 REM — terminar matriz de combate 'A'
0600 FOR X = 1 TO 4:A(1,X) = B(P,X):NEXT X
0610 A(1,5) = 6:A(2,5) = 6:REM — carregadores cheios
0620 A(1,6) = P:A(2,6) = O:REM — Indicador de nomes
0630 REM — Rotina de luta
0640 F = 0:GOSUB 3000
0650 REM — Flag F = 1 quando fatal para o jogador
0660 IF F = 0 THEN 680
0670 P1 = P1–1:IF P1 = 0 THEN END
0680 P = P + 1:IF P< = 6 THEN 210
0690 GOSUB 5100
0700 REM — Já jogaram todos
0710 IF P>1 THEN 740
0720 PRINT "Parabéns — Ganhou o jogo!":END
0730 REM — Pôr 2 jogadores frente a frente
0740 PRINT
0750 X = INT(RND(0)*6 + 1):IF B(X,1) = 0 THEN 750
```



```
0760 Y = INT(RND(0)*6 + 1):IF B(Y,1) = 0 THEN 760
0770 IF X = Y THEN 760
0780 PRINT N$(X);" e ":N$(Y);" discutiram a propósito do ";
0790 PRINT "  gado. Estão à distância de ";
0800 R = INT(RND(0)*200 + 1):PRINT R;" jardas e é
impossível ";
0810 PRINT "  evitar o duelo.":F = 0
0820 FOR A = 1TO 4:A(1,A) = B(X,A):NEXT A
0830 FOR A = 1 TO 4:A(2,A) = B(Y,A):NEXT A
0840 A(1,5) = 6:A(2,5) = 6
0850 A(1,6) = X:A(2,6) = Y
0860 GOSUB 3000
0870 IF F = 0 THEN 900
0880 P1 = P1–1: IF P1 = 1 THEN 0720
0890 REM — Substituir jogadores eliminados?
0900 IF P1 = 6 THEN 200
0910 INPUT "Alguém quer entrar em jogo (S–N)",Q$
0920 IF Q$ = "N" THEN 200
0930 FOR X = 1 TO 6:IF B(X,1)<>0 THEN 960
0940 PRINT "Por favor ";:GOSUB 5000
0950 IF N$(X) = "" THEN 200
0960 NEXT X:GOTO 200
2999 REM — Rotina de combate ********
3000 S = 2:U = 1:T = 2
3030 REM — U = activo (utilizador); T =  alvo
3040 IF A(U,6)< = 6 THEN 3080
3050 REM — Escolha de movimentos pelo computador
3060 GOSUB 5140:GOTO 3090
3070 REM — Escolha de movimentos pelo jogador
3080 GOSUB 5330
3090 ON A + 1 GOTO 3810,3100,3510
3100 REM — Uso das armas
3110 PRINT " levanta ";
3120 IF W$ = "C" PRINT "o revolver ";
3130 IF W$ = "R" PRINT "a espingarda ";
3135 IF S < = 0 THEN 3190
3140 REM — determinar mais rápido
3150  IF A(U,2)>A(T,2) THEN 3180
3160 X = U:U = T:T = X
3170 PRINT ". ";N$(A(U,6)):" é mais rápido a sacar ";
```

```
3180 REM — calcular possibilidade de acertar
3190 S=S-1:H=INT(A(U,3)-SQR(R)-RND(0)*5)
3200 IF W$"R" THEN H=H+10:REM Melhor para
espingarda
3210 A(U,5)=A(U,5)-1:REM Gastar munições
3220 IF RND*100<H THEN 3240:REM — atinge?
3230 PRINT "mas falha.":GOTO 3460
3240 PRINT "e acerta em  ";N$(A(T,6));
3250 W=INT(RND(0)*9+1)
3260 ON W GOTO 3270,3280,3290,3300,3310,3320,3330,
3340,3350
3270 PRINT " no braço esquerdo.":GOTO 3360
3280 PRINT " no braço direito.":GOTO 3360
3290 PRINT " no ombro esquerdo.":GOTO 3360
3300 PRINT " no ombro direito.":GOTO 3360
3310 PRINT " na perna esquerda.":GOTO 3360
3320 PRINT " na perna direita.":GOTO 3360
3330 PRINT " no peito.":GOTO 3360
3340 PRINT " no estômago.":GOTO 3360
3350 PRINT " na cabeça.":GOTO 3420
3360 REM — Efeito do ferimento na força
3370 M=6:IF W$="R" THEN M=4:REM — Maior para
resolver
3380 A(T,4)=A(T,4)-W*M
3390 REM Informar estado do jogador
3400 IF A(T,6)<=6 THEN PRINT A(T,4); " — força
restante."
3410 IF A(T,4) 0 THEN 3470: REM — sobreviveu?
3420 PRINT N$(A(T,6));" está morto!"
3430 REM — Flag e eliminar se for jogador
3440 IF A(T,6)>6 THEN 3680
3450 F=1:B(A(T,6))=0:GOTO 3680
3460 REM — Primeiro dar dois tiros
3470 IF S>0 THEN PRINT "Dispara de novo ";:GOTO
3190
3480 GOTO 3810
3510 REM — Corpo a corpo
3520 W=INT(RND(0)*9+1)
3530 ON W GOTO 3540,3550,3560,3570,3580,3590,3600,
3610,3620
```



```
3540 PRINT "falhou!":GOTO 3810
3550 PRINT "prendeu o braço!":GOTO 3630
3560 PRINT "caiu ao chão!":GOTO 3630
3570 PRINT "meteu um joelho nas costelas!":GOTO 3630
3580 PRINT "apertou o pescoço!":GOTO 3630
3590 PRINT "pontapeou no estômago!":GOTO 3630
3600 PRINT "pontapeou na perna!":GOTO 3630
3610 PRINT "deu um murro no alto da cabeça!":
GOTO 3630
3620 PRINT "tentou estrangular!"
3630 A(T,4)=A(T,4)-INT(A(U,4)*W/10):REM — Ajustar
força
3640 IF A(T,4)<=0 THEN 3670:REM — KO?
3650 IF A(T,6)<=6 THEN PRINT A(T,4):" força restan
te"
3660 GOTO 3810
3670 PRINT N$(A(T,6)):" está K.O.!"
3680 REM — Ajustamento se o vencedor for o jogador
3690 IF A(U,6)>6 RETURN
3700 V=A(U,6):PRINT "Boa luta, ";N$(V);"!"
3710 B(V,5)=B(V,5)+I+INT(RND(0)*20/L+1)
3720 B(V,1)=B(V,1)+INT(RND(0)*5+5
3730 REM — Ajustar outros coeficientes?
3740 IF B(V,1)<90 THEN 3790
3750 FOR X=2 TO 4
3760 B(V,X)=B(V,X)+INT((100-B(V,X))/5*RND(0))+1
3765 REM Não permitir invencibilidade
3770 IF B(V,X)>=100 THEN B(V,X)=100-INT(RND
(0)+*10+1)
3780 NEXT X:IF B(V,1)>=100 THEN B(V,1)=100-INT
(RND(0)*10+1)
3790 X=V:GOSUB 5040:RETURN
3800 REM — Movimentos alternados
3810 X=T:T=U:U=X:GOTO 3040
4999 REM — Entrada de caracteres ******
5000 INPUT "Escreva o seu nome ",N$(X)
5010 IF N$(X)="" THEN RETURN
5020 B(X,5)=0:REM — Reputação
5030 FOR Y=1 TO 4:B(X,Y)=INT(RND(0)*90+10):NEXT
Y
5040 PRINT "Os seus coeficientes são —"
```

```
5050 PRINT "Bravura – ";B(X,1);"%"
5060 PRINT "Velocidade – ";B(X,2);"%"
5070 PRINT "Pontaria — ";B(X,3);"%"
5080 PRINT "Força — ";B(X,4);"%"
5090 PRINT "Reputação — ";B(X,5):RETURN
5100 PRINT
5110 PRINT "****** Por favor carregue em
RETURN ******"
5120 INPUT Q$:PRINT:RETURN
5130 REM — Movimento do fora-da-lei *****
5140 PRINT :PRINT N$(0);:A=0
5150 IF A(U,5)<>0 THEN 5170
5155 REM — Dois tempos para recarregar
5160 PRINT" carrega a arma.":A(U,5)=999:GOTO 5280
5170 IF A(U,5)<>999 THEN 5190
5180 PRINT " está a carregar a arma.":A(U,5)=6:GOTO
5280
5190 IF A(U,3)>60 THEN 5260
5200 IF R>=3 THEN 5210
5205 PRINT "ataca-o - ";:A=2:RETURN
5210 IF RND>.25 THEN 5260
5220 PRINT " avança. A distância é agora ";
5225 IF R=0 THEN PRINT "Ele ";:GOTO 5205
5230 GOSUB 5290:RETURN
5240 REM — Flag de ataque 'A'=0=Não há ataque
5250 REM — A=1=Armas e A=2=Luta corpo a corpo
5260 A=1:W$="R":IF R>100 GOTO 5280
5270 IF RND>.33 W$="C"
5280 RETURN
5290 REM — Avançar
5300 R=INT(R−5−(RND(0)*R/4)):IF R<0 THEN R=0
5310 PRINT R;" jardas."
5320 RETURN
5330 REM — Escolha de movimentos pelo jogador
5340 PRINT :A=0:PRINT N$(A(U,6));" — ";
5350 IF A(U,5)<>999 THEN 5370
5360 PRINT "Você está a carregar a arma.":A(U,5)=6:
RETURN
5370 INPUT "Carregar, disparar, mover ou lutar
(C,D,M,L)",Q$
```



```
5380 IF Q$<>"D" THEN 5470
5390 IF A(U,5)>0 THEN 5410
5400 PRINT "Disparou 6 tiros — tem de carregar a
arma.":GOTO 5350
5410 INPUT "Revólver ou espingarda (R−E)",W$
5420 IF W$="E"THEN A=1:PRINT N$(A(U,6));:
RETURN
5430 IF R<100 THEN 5450
5440 PRINT "A distância é demasiado grande para um
revólver.":GOTO 5370
5450 IF W$<>"R" THEN PRINT "Por favor escreva R ou
E":GOTO 5410
5460 A=1:PRINT N$(A(U,6));:RETURN
5470 IF Q$="C" THEN A(U,5)=999:RETURN
5480 IF Q$<>"M" THEN 5510
5485 IFR=0 THEN 5540
5490 PRINT "Você avança. A distância é agora ";:GOSUB
5290
5495 IF R=0 THEN 5540
5500 RETURN
5510 IF Q$<>"L" PRINT "Por favor escreva C,D,M ou
L":GOTO 5370
5520 IF R<3 THEN 5540
5530 PRINT "Está demasiado longe para atacar.":GOTO
5370
5540 A=2:PRINT "Você ataca-o — ";:RETURN
9000 PRINT "Neste jogo podem participar 6 pessoas ";
9010 PRINT "representando caracteres do Far-West. Encon
tram ";
9020 PRINT " foras-da-lei, desafiam-nos para lutas a tiro ";
9030 PRINT "e corpo a corpo. Os jogadores ganham pontos ";
9040 PRINT " de reputação, vencendo aquele que primeiro ";
9050 PRINT "atingir 500, ou o único que ficar vivo."
9060 PRINT :INPUT "Carregue em RETURN para
começar",Q$
9070 PRINT "Ao ser-lhe perguntado, cada jogador deve ";
9080 PRINT " escrever o seu nome. Quando não há mais jo
gadores ";
9090 PRINT "carrega-se em RETURN sem escrever
qualquer ";
```

```
9100 PRINT " nome. Os jogadores recebem um valor de
bravura,";
9110 PRINT " velocidade e pontaria com a arma, e
força. ";
9115 PRINT "Deve tomar nota destes valores, pois
ajudá-";
9120 PRINT "lo-ão a tomar algumas decisões."
9130 PRINT "Todos os jogadores possuem um revólver
.45 ";
9140 PRINT " e uma espingarda de repetição Remington."
9150 PRINT :INPUT "Carregue em RETURN",Q$
9160 PRINT "Os revólveres produzem mais danos, mas
não ";
9170 PRINT " são tão rigorosos como a espingarda, e só ";
9180 PRINT "disparam até distâncias de 100 jardas. ";
9185 PRINT "Nenhum jogador pode disparar mais de 6
balas ";
9190 PRINT "sem recarregar, mas o mais rápido
a tirar a "
9195 PRINT "arma dispara duas vezes primeiro. "
9200 PRINT "Em vez de disparar, pode aproximar-se do seu ";
9210 PRINT " inimigo a fim de aumentar as possibilidades "
9220 PRINT " de acertar nele, e se se aproximar
suficientemente ";
9230 PRINT " pode lutar corpo a corpo sem usar as
armas. "
9240 PRINT "Se fugir do seu inimigo, isto diminui os seus ";
9245 PRINT " pontos de bravura. Se estes atingirem zero
será ";
9250 PRINT " retirado de jogo."
9260 PRINT :INPUT "Carregue em RETURN",Q$
9270 PRINT "Quando um jogador é morto, pode ser
substituído ";
9280 PRINT " por outro. Por vezes dois jogadores
enfrentar-";
9290 PRINT " se-ão entre si, aplicando-se as mesmas regras."
9300 PRINT "E é tudo. Bebam um copo, e partamos..."
9310 PRINT :INPUT "Carregue em RETURN",Q$:
RETURN
```


Exemplo de execução

```
Deseja instruções (S—N)? N

Jogador n. 1 — Escreva o seu nome? TINY TIM
Os seus coeficientes são —
Bravura — 35%
Velocidade — 32%
Pontaria — 98%
Força — 10%
Reputação — 0

Jogador n. 2 — Escreva o seu nome? DANGEROUS DAN
Os seus coeficientes são —
Bravura — 67%
Velocidade 19%
Pontaria — 80%
Força — 54%
Reputação — 0

Jogador n. 3 — Escreva o seu nome? CLINT EASTWOOD
Os seus coeficientes são —
Bravura — 92%
Velocidade — 59%
Pontaria — 48%
Força — 41%
Reputação — 0

Jogador n. 4 — Escreva o seu nome? SAM SMALL
Os seus coeficientes são —
Bravura 73%
Velocidade — 73%
Pontaria — 73%
Força — 80%
Reputação — 0

Jogador n.5 — Escreva o seu nome?
****** Por favor carregue em RETURN ******
TINY TIM — encontrou Mexican Pete fora da cidade.
Ele exige que lhe dê o seu dinheiro.
```

Está à distância de 64 jardas.
Foge ou luta (F–L)? L

TINY TIM — Carregar, disparar, mover ou lutar (C,D,M,L)? D
Revólver ou espingarda (R–E)? R
TINY TIM levanta o revólver. Mexican Pete é mais rápido a sacar e acerta em TINY TIM no braço esquerdo.
4 — força restante.
Dispara de novo e acerta em TINY TIM no peito.
— 38 — força restante
TINY TIM está morto!
****** Por favor carregue em RETURN ******

DANGEROUS DAN — encontrou Jake the Halfbreed fora da cidade.
Ele exige o seu dinheiro.
Está à distância de 62 jardas.
Foge ou luta (F–L)? L

DANGEROUS DAN — Carregar, disparar, mover ou lutar (C,D,M,L)? D
Revólver ou espingarda (R–E)? R
DANGEROUS DAN levanta o revólver. Jake the Halfbreed é mais rápido
rápido a sacar e acerta em DANGEROUS DAN no estômago.
6 — força restante.
Dispara de novo e acerta em DANGEROUS DAN na cabeça.
DANGEROUS DAN está morto!

****** Por favor carregue em RETURN ******

CLINT EASTWOOD — encontrou Mexican Pete num desfiladeiro.
Ele exige o seu dinheiro.
Está à distância de 136 jardas.
Foge ou luta (F–L)? L

CLINT EASTWOOD — Carregar, disparar, mover ou lutar (C,D,M,L,)? M
Você avança. A distância é agora 112 jardas.

Mexican Pete levanta a espingarda. CLINT EASTWOOD é mais rápido a sacar e acerta em Mexican Pete na perna direita.
Mexican Pete está morto!
Boa luta, CLINT EASTWOOD!
Os seus coeficientes são —
Bravura — 97%
Velocidade — 67%
Pontaria — 56%
Força — 45%
Reputação — 10

****** Por favor carregue em RETURN ******

SAM SMALL — encontra Pedro Gonzalez na estação de caminho de ferro.
Ele diz que você é um cobarde.
Está à distância de 35 jardas.
Foge ou luta (F–L)? L

SAM SMALL — Carregar, disparar, mover ou lutar (C,D,M,L)? M
Você avança. A distância é agora 27 jardas.
Pedro Gonzalez levanta o revólver. SAM SMALL é mais rápido a sacar mas falha.
Dispara de novo e acerta em Pedro Gonzalez no estômago.
Pedro Gonzalez está morto!
Boa luta, SAM SMALL!
Os seus coeficientes são —
Bravura — 82%
Velocidade — 73%
Pontaria — 73%
Força — 80%
Reputação — 12

****** Por favor carregue em RETURN ******

SAM SMALL e CLINT EASTWOOD discutiram a propósito do gado.
Estão à distância de 120 jardas e é impossível evitar o duelo.

SAM SMALL — Carregar, disparar, mover ou lutar (C,D,M,L)? M
Você avança. A distância é agora 88 jardas.

CLINT EASTWOOD — Carregar, disparar, mover ou lutar (C,D,M,L)? M
Você avança. A distância é agora 79 jardas.

SAM SMALL — Carregar, disparar, mover ou lutar (C,D,M,L)? M
Você avança. A distância é agora 72 jardas.

CLINT EASTWOOD — Carregar, disparar, mover ou lutar (C,D,M,L)? M
Você avança. A distância é agora 60 jardas.

SAM SMALL — Carregar, disparar, mover ou lutar (C,D,M,L)? M
Você avança. A distância é agora 43 jardas.

CLINT EASTWOOD — Carregar, disparar, mover ou lutar (C,D,M,L)? M
Você avança. A distância é agora 34 jardas.

SAM SMALL — Carregar, disparar, mover ou lutar (C,D,M,L)? M
Você avança. A distância é agora 25 jardas.

CLINT EASTWOOD — Carregar, disparar, mover ou lutar (C,D,M,L)? D
Revólver ou espingarda (R–E)? R
CLINT EASTWOOD levanta o revólver. SAM SMALL é mais rápido a sacar e acerta em CLINT EASTWOOD no estômago.
–9 — força restante.
CLINT EASTWOOD está morto!
Boa luta, SAM SMALL!
Os seus coeficientes são —
Bravura — 90%
Velocidade — 70%

Pontaria — 77%
Força — 83%
Reputação — 19
Parabéns — Ganhou o jogo!

READY
*

XXVI

BOND 007 (13 K)

Eis a sua possibilidade de superar James Bond! A sua tarefa consiste em recuperar certos planos secretos e entregá-los a M, em Londres. Só três pessoas conhecem a localização desses planos — e uma delas tem-nos em seu poder. O problema consiste em que você tem de interrogar 100 suspeitos em 10 cidades diferentes; e pior ainda, nem todos eles lhe vão dizer a verdade — alguns tentarão pelo contrário obstruir a sua investigação. Vai descobrir como é difícil esta profissão, mesmo tendo licença para matar...

O jogo baseia-se numa sequência de ligações. Um agente tem os planos em seu poder, e dois outros conhecem este facto. Seis outros sabem quem são os três primeiros, mas não sabem qual deles tem os planos; sabem quem sabe... Dezasseis outros (ainda me está a acompanhar?!) julgam que sabem quem sabe, e os outros ouviram rumores ou não sabem coisa alguma (não quer descansar um pouco?).

De qualquer modo, se tudo se resumisse a isto (!), bastaria alguma persistência para chegar aos planos. Ora acontece que os agentes podem ser homens ou mulheres. Pior ainda, podem ser ingleses, americanos, russos ou locais. Os agentes ingleses dirão normalmente a verdade, e os americanos indicarão o agente que convém contactar mas falsearão o nome da cidade. Os agentes locais dir-lhe-ão qual a cidade correcta mas indicarão um agente errado, e como é sabido não vale a pena acreditar numa palavra do que disser um agente russo. Outro inconveniente ainda: você nem sempre saberá qual a nacionalidade do agente, e portanto não saberá se está a mentir ou não! A única excepção a isto são os agentes ingleses a quem você fornecer a senha correcta.

Ainda pensa que consegue encontrar os planos?

Para ajudar, o agente inglês residente em cada cidade conhece os números de alguns dos contactos na cidade (nem todos, pois podem estar clandestinos). Por outro lado, se amedrontar alguns agentes suficientemente, dir-lhe-ão a verdade. Outros não mentirão se lhes der uma tareia, enquanto alguns agentes (femininos) estarão dispostos a responder às suas perguntas se gostarem de si — nem tudo é mau!

Pode conseguir alguns indícios a partir da mortandade que entretanto prossegue: aparecem corpos a flutuar no Tibre, caindo do muro de Berlim ou decorando as sarjetas de Nova Iorque. Estas mortes acabam por ser conhecidas por "M", que lhe passa as informações. Gradualmente você acaba por descobrir em quem pode confiar ou não. A pouco e pouco são cada vez mais os indícios que apontam numa dada direcção...

Tenha cuidado! Existem muitos outros pormenores neste jogo que seria demasiado longo explicar-lhe. É muito melhor ler a listagem e jogar, mas não espere ganhar — pelo menos nos primeiros tempos...

O jogo evolui em torno da matriz de espiões A, na qual para cada um de cem agentes existem quatro colunas. A primeira é o sexo do agente: 0 para as mulheres, 1 para os homens. A coluna seguinte indica a localização do agente, um número entre 1 e 10 que corresponde às dez cidades do quadro L$. A terceira indica a pessoa que o agente julga possuir os planos, enquanto a quarta indica a nacionalidade do agente. A coluna 3 produz as ligações entre os agentes, mas não se esqueça de que algumas destas serão enganadoras e outras não conduzirão a nada; isto é, a coluna estará em branco.

Chamo em especial a sua atenção para as linhas 110 e 130, que se tornam necessárias devido ao modo como o computador escolhe um número aleatório: não escolhe. Para pôr a questão mais claramente, quando se pede ao computador que produza um número aleatório, aquele realiza um cálculo a partir de um número anterior. Isto significa que, a partir de um dado número, o computador indicará, produzirá sempre um mesmo número! Como a maior parte das versões da BASIC partirão do mesmo número original sempre que são ligadas, segue-se que a partir daí apresentarão sempre a mesma sequência de números. Como no entanto a sequência só será repetida muitos milhares de números

depois, chama-se de facto ao número produzido "pseudo-aleatório".

Nestas condições, de cada vez que ligar o computador e carregar um dado programa irá jogar o mesmo jogo, porque as instruções de produção de números aleatórios apresentarão sempre os mesmos números. É por isso que utilizo aqui as linhas 110 e 130. Qualquer caracter escrito é transformado no seu código ASCII e o computador produz esse número de valores aleatórios. Claro que se for sempre escrito o mesmo caracter o jogo se repetirá, mas é improvável que tal aconteça; de qualquer modo, a questão só é importante da primeira vez que o jogo é jogado depois de ligar a máquina.

A linha 480 produz uma palavra código de 3 letras recorrendo a três números aleatórios equivalentes aos códigos ASCII entre A e Z, juntando depois estas. Se quiser pode omitir as linhas 540 a 600 inclusive; servem apenas para imprimir em papel os números 1 a 100 dispostos em colunas com títulos (a minha impressora está ligada ao "port" 7). Esta lista serve apenas para facilitar o jogo, podendo ser substituída por outra feita à mão num pedaço de papel.

Você "contacta" um agente escrevendo muito simplesmente o seu número — desde que estejam ambos na mesma cidade. A resposta será então imediata. Se não acreditar no que lhe dizem, pode atacar um agente; mas note que isto lhe custa 200 pontos, diminuídos à sua pontuação final se conseguir descobrir os planos (se não conseguir a sua pontuação é zero). Por outro lado, se o agente o ataca, a sua pontuação não é afectada.

Se o agente contactado é inglês, é-lhe pedida a sua senha. Esta é-lhe indicada no início do jogo, e se a tiver esquecido ou se enganar o agente dar-lhe-á indicações erradas.

Em conclusão, este jogo não é fácil — nem pretende sê-lo. Tentei incluir tantos aspectos "reais" quanto me foi possível, e isso contribui evidentemente para dificultá-lo.

Lista de variáveis

Ax(X,Y)	Matriz dos espiões, onde X = Número do agente e Y = 1 — Sexo 0 ou 1 (99 se morreu) 2 — nacionalidade 0-4 3 — Informação 0-100 4 — Localização 1-10
Q(X)	Quadro de ligações temporárias
L$(X)	Quadro contendo nomes de cidades
Q$	String de entradas
Q	Contador de séries aleatórias
X	Contador geral
P	Agente com planos
Y	Contador geral
Z	Contador geral; também atributo do jogador actual
A	Resposta do agente
B	Contador geral
L	Posição actual
S	Pontuação actual
L1	Contador de posições
R	Número aleatório; também número de resposta
C1	Número de contacto
C	Contacto actual
P$	Palavra de passe
F	Flag

Listagem do programa

```
0010 PRINT TAB(20); "BOND 007"
0020 PRINT TAB(20);"= = = = ="
0030 PRINT
0040 DIM A(100,4),Q(8), L$(10)
0050 DEF FNA(X) = INT (RND(0)*100 + 1)
0060 DEF FNB(X) = INT(RND(0)*4 + 1)
0070 DEF FNC(X) = INT(RND(0)*10 + 1)
0090 INPUT "Deseja instruções",Q $
0100 IF LEFT$(Q$,1) = "S" THEN GOSUB 8000
0110 INPUT "Importa-se de escrever um caracter", Q$
```

```
0120 PRINT "Um momento."
0130 Q=ASC(Q$):FOR X=1 TO Q:A=RND:NEXT X
0140 REM — Definir matriz
0160 DATA Londres, Paris, Roma, Nova Iorque, Tóquio
0170 DATA Istambul, Hong-Kong, Caracas, Berlim,
Moscovo .
0180 FOR X=1 TO 10:READ L$(X):NEXT X
0220 REM — Colocar planos secretos
0230 P=FNA(X):A(P,3)=P
0240 REM — Dois sabem onde estão
0250 Z=3:FOR X=1 TO 2:GOSUB 9000
0260 Q(X)=A:A(A,3)=P:NEXT X
0270 REM — Seis sabem quem sabe
0280 FOR B=1 TO 2:FOR Y=1 TO 3:GOSUB 9000
0290 A(A,3)=Q(B):X=X+1:Q(X)=A:NEXT Y: NEXT B
0300 REM — 16 pensam que sabem
0310 FOR Y=1 TO 2:FOR X=1 TO 8:GOSUB 9000
0320 A(A,3)=Q(X):NEXT X:NEXT Y
0340 FOR X=1 TO 100:IF RND>.8 THEN 360
0350 IF A(X,3)=0 THEN A(X,3)=FNA(X)
0360 A(X,4)=FNC(X):REM — Posição
0370 IF RND .1 THEN A(X,1)=1:REM sexo
0380 IF RND .3 THEN A(X,2)=FNB(Y):REM —
Nacionalidade
0390 NEXT X
0450 PRINT " 007 — descubra esses planos e traga-mos!"
0460 PRINT "Sugiro que contacte o nosso agente n.";
0470 FOR A=1 TO 100:IF A(A,2)<>1 THEN NEXT A
0475 PRINT A;" em ";L$(A(A,4));" — parece estar ";
0480 FOR X=1 TO 3:P$=P$+CHR$(INT(RND(0)*
26+65)):NEXT X
0490 PRINT "em perigo. A senha é ";P$;"."
0510 PRINT "Oh — e alguns dos agentes em Londres neste ";
0520 PRINT " momento (ingleses e estrangeiros) são — ";
0530 L=1:Z=1:S=10000:GOSUB 9540
0540 INPUT "Deseja uma lista impressa",Q$
0550 IF LEFT$(Q$,1)<>"S" THEN 1000
0560 INPUT "Tem a impressora ligada",Q$
0570 IF LEFT$(Q$,1)<>"S" THEN 1000
0580 LINE=100:FOR X=1 TO 4
```



```
0590 PRINT #7, "AGENTE LOCAL INDÍCIO";
0600 PRINT #7,TAB(X*20);:NEXT:PRINT #7
0610 FOR X=1 TO 25
0620 FOR Y=0 TO 3
0630 PRINT #7,TAB(20*7);X+25*Y;
0640 NEXT Y:PRINT #7
0650 NEXT X:LINE=72
0999 REM Início do loop principal
1000 GOSUB 9020:GOTO 1200
1010 INPUT "Para onde, James",Q$
1020 FOR L1=1 TO 10:IF Q$=L$(L1) THEN 1070
1030 NEXT L1
1040 PRINT "Não existe esse local no jogo."
1050 PRINT "Não deve ter escrito bem...":S=S-10:GOTO
1010
1070 IF L1=L THEN 1130
1080 IF RND(0)>.1 THEN 1130
1090 PRINT "O avião foi sabotado e explodiu em voo! "
1100 IF RND(0)<.4 THEN 1640
1120 PRINT "No entanto, você não ia nele!"
1130 PRINT "Você está agora em ";L$(L1:L=L1:S=S-100
1140 PRINT :IF RND>.1 THEN 1200
1150 PRINT :PRINT "Há uma mensagem de "M", vinda de
Londres."
1160 R=FNA(X):IF A(R,1)>10 THEN 1160
1170 A(R,1)=99:Z=A(R,4):PRINT "A mensagem diz —"
1180 PRINT "O AGENTE":GOSUB 9570:PRINT "N. ";R;
" EM ";L$(Z);
1190 PRINT " FOI ASSASSINADO."
1192 IF P<>R THEN PRINT " Fim da mensagem.":GOTO
1200
1194 PRINT " TINHA OS PLANOS NO BOLSO. VOLTE
PARA LONDRES.":GOTO 1640
1200 PRINT :PRINT "Você está ainda em ";L$(L)
1210 INPUT "O seu contacto ",C1:S=S-2:IF C1=0 THEN
1010
1220 IF C1>100 THEN PRINT "Número muito alto!":
GOTO 1210
1230 IF C=C1 THEN GOSUB 9340:S=S-5:PRINT
"acabou de sair.":GOTO 1140
```

```
1240 IF A(C1,1)>10 THEN GOSUB 9340:PRINT “está morto!”:GOTO 1270
1250 C=C1:IF A(C,4)=L THEN 1300
1260 GOSUB 9340:PRINT “Não está aqui!”
1270 PRINT “Você vê dois polícias aproximando-se.”
1280 IF RND>.1 THEN 1290
1285 GOSUB 9050:GOTO 1140
1290 PRINT “Consegue escapar pelas traseiras.”:GOTO 1140
1300 PRINT :PRINT:GOSUB 9630
1302 IF A(C,2)=1 THEN 1900
1304 IF RND>.5 THEN 1530
1305 ON A(C,2)+1 GOTO 1310,1360,1320,1360,1340
1310 R=50:GOTO 1330
1320 R=5
1330 IF R<=RND(0)*100 THEN 1360
1340 IF RND>.8 THEN 1380
1341 R=FNA(X):IF R=A(C,3) THEN 1341
1345 IF R=C THEN 1341
1350 GOTO 1370
1360 R=A(C,3)
1370 IF R>0 THEN 1390
1380 PRINT “‘Nada sei, sr. Bond — é verdade!’”:GOTO 1490
1390 IF R=P THEN 1340
1400 PRINT “‘Procure o agente n.”;R;“ em ”;
1410 ON A(C,2)+1 GOTO 1420,1470,1430,1450,1450
1420 L1=50:GOTO 1440
1430 L1=5
1440 IF L1<=RND(0)*100 THEN 1470
1450 L1=FNC(X):IF L1=L THEN 1450
1460 GOTO 1480
1470 L1=A(R,4)
1480 PRINT L$(L1);“’”:IF A(C,2)=1 THEN 1200
1490 PRINT :INPUT “Acredita”,Q$
1500 IF LEFT$(Q$,1)=“s” then 1140
1510 IF A(C,1)=1 THEN 1515
1512 GOSUB 9350:GOTO 1630
1515 S=S-200:REM Castigo por provocação
1520 PRINT “Você ataca-o!”:GOSUB 9300:GOTO 1630
```



```
1530 REM — Segundo interrogatório
1540 A=INT(A(C,1)*A(C,2)*L*RND(0))+1
1545 IF A>7 THEN 1270
1550 ON A GOTO 1560,1380,1340,1800,1610,1710,1270
1560 IF A(C,1)=0 THEN 1700
1570 GOTO 1800
1610 PRINT “ De súbito, ”;:GOSUB 9340:PRINT “ ataca-o!”
1620 GOSUB 9300
1630 ON F+1 GOTO 1700,1650,1640,1340
1640 PRINT “Perdeu! Era ”;P;“ quem tinha os planos.”:END
1650 IF L>=9 THEN GOSUB 9130
1660 PRINT “Acordou no meio do lixo de um beco. Está a ”;
1670 PRINT “ sangrar e tem o corpo cheio de nódoas negras.”
1690 A(C,4)=FNC(X):S=S-250:GOTO 1140   Fig. 163
1700 PRINT “ Está bem! diz o agente n.”;C;“.”
1710 IF C<>A(C,3) THEN 1750
1720 PRINT “‘Tenho os planos aqui!’”
1730 PRINT: PRINT :PRINT “VOCÊ GANHOU!”
1740 PRINT “A sua pontuação foi ”;S:END
1750 IF A(C,2)<>4 THEN 1770
1760 IF RND .1 THEN 1340
1770 IF A(C,3)=0 THEN 1380
1780 PRINT “‘Contacte agente n.”;A(C,3);“em”;L$-(A(A(C,3),4));“’”
1790 GOTO 1140
1800 PRINT “De súbito, ”;:GOSUB 9340:PRINT “Corre. Você ”;
1810 PRINT “persegue ”;:GOSUB 9780
1820 IF RND>.3 THEN 1870
1830 PRINT “mas perde ”;:GOSUB 9780:PRINT “ num labirinto de ruelas.”
1840 PRINT “Maldita sorte! Você sente que ”;
1850 PRINT “ não passará muito tempo antes de ”;:GOSUB 9340:PRINT “ fugir de avião.”
1860 A(C,4)=FNC(X):S=S-50:GOTO 1140
1870 PRINT “e encontra-o num parque de estacionamento.”:GOTO 1700
```

```
1900 REM — Agente inglês
1910 PRINT :INPUT "Qual é a senha",Q$
1920 IF P$<>Q$ THEN 1310
1925 IF L=1 THEN 1950
1930 PRINT :PRINT "Bem-vindo a ";L$(L); " sr. Bond.";
1940 PRINT "Londres disse-me que estava para chegar."
1950 IF C=A (C,3) THEN 1720
1960 INPUT "Deseja uma lista dos agentes que estão
aqui",Q$
1970 IF LEFT$(Q$,1)="S" THEN Z=L:GOSUB 9540
1980 IF A(C,3) 0 THEN 1360
2010 PRINT "Nada ouvi dizer sobre esses planos — ";
2020 PRINT " nem um rumor!":GOTO 1140
8000 PRINT "Neste jogo você substitui James Bond,
agente ";
8010 PRINT " secreto inglês 007. A sua tarefa consiste
em ";
8020 PRINT "descobrir onde se encontram uns planos
secretos ";
8030 PRINT "roubados a uma potência estrangeira. Para
tal ";
8040 PRINT "necessitará de usar uma combinação de
inteligência, ";
8059 PRINT " força e sorte."
8060 PRINT " Existem 100 agentes secretos suspeitos a ";
8070 PRINT "considerar. São ingleses, americanos, russos, ";
8080 PRINT " locais e neutros, estando espalhados por 10
cidades."
8090 PRINT " Convirá usar um lápis e papel para tomar
notas. ";
8100 PRINT :INPUT "Carregue em RETURN quando
estiver pronto",Q$:PRINT
8110 PRINT :PRINT "Para interrogar um agente suspeito,
deve ";
8120 PRINT " obviamente encontrar-se na mesma cidade.
Os ";
8130 PRINT " agentes podem dizer a verdade ou mentir.
Os ";
8140 PRINT " agentes russos mentirão sempre, e os
ingleses ";
```



```
8150 PRINT " dir-me-ão sempre a verdade. Os outros
podem fazer ";
8160 PRINT " ambas as coisas. De início não saberá que
tipo ";
8165 PRINT "de agente encontra, nem se está ou não a dizer
a";
8170 PRINT " verdade. Se não acredita nele pode
atacá-lo ";
8180 PRINT "(a menos que se trate de uma senhora), mas ";
8190 PRINT " tenha cuidado — pode perder e ser
morto!. Qualquer ";
8200 PRINT " agente que o agredir partirá imediatamente
para ";
8210 PRINT " outra cidade."
8220 PRINT :INPUT"Carregue em RETURN",Q$
8222 PRINT :PRINT "Para interrogar um agente, escreva
penas ";
8224 PRINT " o seu número. Para mudar de cidade,
escreva 0. "
8230 PRINT " Não pode interrogar o mesmo suspeito ";
8240 PRINT "duas vezes de seguida, mas pode voltar mais
tarde. ";
8250 PRINT " Se procurar alguém que não vive numa dada ";
8260 PRINT " cidade, a polícia interessa-se por si: tenha ";
8270 PRINT "portanto cuidado! Encontrará ainda outras ";
8280 PRINT " surpresas, mas se mantiver a calma
conseguirá ";
8290 PRINT " descobrir a verdade e encontrar os planos ";
8300 PRINT " roubados. Boa sorte!"
8310 PRINT
9000 A=FNA(X):IF A(A,Z)<>0 THEN 9000
9010 RETURN
9020 PRINT
9030 PRINT "****** Carregue em RETURN ******"
9040 INPUT Q$:PRINT:PRINT
9050 PRINT "Saem polícias de trás dos tapumes!"
9060 IF L<=8 THEN 9080
9070 PRINT "Não se esqueça de que está num país
comunista."
9080 INPUT "Resiste à prisão (S-N)?",Q$
```

```
9090 IF LEFT$(Q$,1) = "N" THEN PRINT "É conduzido a
";:GOTO 9140
9100 GOSUB 9720:ON F + 1 GOTO 9120,9130,9290
9120 PRINT "Má sorte!":GOTO 1640
9130 PRINT "Quando acorda, vê-se ";
9140 PRINT "numa cela nua.":GOSUB 9020
9150 PRINT "Entra um guarda. ";
9160 IF L< = THEN 9270
9170 ON FNB(X) GOTO 9180,9200,9210,9210
9180 PRINT "Estou a trabalhar com os serviços secretos ";
9185 PRINT "  ingleses,  ":GOTO 9240
9200 PRINT "'Trabalho para a CIA', ";:GOTO 9240
9210 PRINT "' Tenho uma mensagem para si, 'de ";
9220 PRINT "  alguém chamado M. Diz que está
despedido!'":GOTO 1640
9240 PRINT "'Vim para levá-lo daqui.'"
9250 PRINT "Algumas horas depois está em local
seguro.":F = 0:RETURN
9270 PRINT "'Vamos!' diz, 'Está ali alguém que gosta de ";
9280 PRINT "  si — acabam de pagar-lhe a fiança!'":F = 0
9290 RETURN
9300 PRINT "A luta continua, usando ambos todas as ";
9301 PRINT "  armas disponíveis — punhos, pontapés,
adagas ";
9302 PRINT "  pistolas, karate, judo. O seu opositor  ";
9303 PRINT "consegue rapidamente sobrepor-se, ";
9310 F = FNA(X):IF F>98 THEN F = 2:PRINT "e mata
-o!":F = 2:RETURN
9320 IF F>90 THEN F = 1:PRINT "e deixa-o
inconsciente!":RETURN
9330 F = 0:PRINT "mas você consegue";
9332 IF A(C,1) = 1 THEN PRINT " dominá-lo.":RETURN
9334 PRINT "dominá-la.":RETURN
9340 IF A(C,1) = 0 THEN PRINT " Ela ";:RETURN
9341 PRINT "ele ";:RETURN
9350 PRINT C; " diz suavemente, 'Os únicos planos que
tenho,";
9352 PRINT "James, pode consultá-los à sua vontade... " ;
9354 PRINT "Ela beija-o arrebatadamente.":INPUT "Você
corresponde",Q$
```



```
9358 IF LEFT$(Q$,1) = "S" THEN 9420
9360 IF A(C,2)< = 2 THEN 9400
9370 PRINT "Ela fica furiosa. Tirando um punhal ";
9380 PRINT "  da meia, ataca-o!"
9390 GOTO 9300
9400 PRINT "Ela suspira. "Você é tolo, James.'"
9410 F = 3:RETURN
9420 IF A(C,2)< = 2 THEN 9500
9430 PRINT "Oh! Ela era uma agente russa. Enquanto você ";
9440 PRINT "  dormia, ela  ";:IF RND>.1 THEN 9480
9460 PRINT "cortou-lhe o pescoço (gentilmente, claro...)"
9470 GOTO 1640
9480 F = 1:PRINT "  deu-lhe amorosamente alguns murros
no peito."
9490 RETURN
9500 PRINT "Na manhã seguinte, ao pequeno-almoço, ela
diz-lhe ";
9510 Z = ABS(A(C,3)):IF C = P THEN 1720
9515 IF A(C,3) = 0 THEN PRINT "  que não sabe nada."
9520 F = 0:RETURN
9540 FOR X = 1 TO 100:IF RND<.1 THEN 9560
9550 IF A(X,4) = Z THEN PRINT X;
9560 NEXT X:PRINT:PRINT:RETURN
9570 ON A(R,2) + 1 GOTO 9580,9590,9600, 9610, 9620
9580 PRINT "DESCONHECIDO";:RETURN
9590 PRINT "INGLÊS";:RETURN
9600 PRINT "AMERICANO";:RETURN
9610 PRINT "LOCAL";:RETURN
9620 PRINT "RUSSO";:RETURN
9630 REM — Descrever agente
9640 PRINT "O agente é ";
9650 IF A(C,1) = 1 THEN 9670
9660 PRINT "  uma jovem elegante e sensual. ":RETURN
9670 IF C = INT(C/3)*3 THEN 9700
9680 IF C = INT(C/3)*3 + 1 THEN 9710
9690 PRINT "  um velhote aparentemente doente.":
RETURN
9700 PRINT "  um respeitável homem de negócios.":
RETURN
```

```
9710 PRINT "  um sujeito de forte constituição.":RETURN
9720 PRINT "Atacam-no por todos os lados.
";:F=L+FNC(X)
9740 IF F=13 THEN F=2:PRINT "Você está morto!":
RETURN
9750 IF F<5 THEN F=0:PRINT "Você foge por uma
janela.":RETURN
9760 F=1:PRINT "Você ficou inconsciente...":RETURN
```

Exemplo de execução

Deseja instruções? Não.
Importa-se de escrever um caracter? 5
Um momento.

007 — Descubra esses planos e traga-mos!
Sugiro que contacte o nosso agente n. 6 em Tóquio — parece estar em perigo. A senha é GBK.
Oh — e alguns dos agentes em Londres neste momento (ingleses e estrangeiros) são 10 17 51 54 64 75 81 82 89
Deseja uma lista impressa? Não

****** Carregue em RETURN ******

Você está ainda em Londres
O seu contacto? 10
O agente é um sujeito de forte constituição
'Procure o agente n. 17 em Nova Iorque'
Acredita? Não
Você ataca-o!
A luta continua, usando ambos todas as armas disponíveis — punhos, pontapés, adagas, pistolas, karate, judo. O seu opositor consegue rapidamente sobrepor-se, mas você consegue dominá-lo.
'Está bem!' diz o agente n. 10. 'Procure o agente n. 39 em Moscovo.'

Você está ainda em Londres.
O seu contacto? 17

O agente é um velhote aparentemente doente.
'Procure o agente 7 em Tóquio.'
Acredita? Sim.

Você está ainda em Londres.
O seu contacto? 39
Não está aqui!
Você vê dois polícias aproximando-se
Consegue escapar pelas traseiras.

Você está ainda em Londres
O seu contacto? 0
Para onde, James? Tóquio.
O avião foi sabotado e explodiu em voo!
No entanto, você não ia nele!
Você está agora em Tóquio
Há uma mensagem de 'M', vinda de Londres.
A mensagem diz —
O AGENTE INGLÊS N. 33 EM CARACAS FOI ASSASSINADO
Fim da mensagem

Você está ainda em Tóquio.
O seu contacto? 6

O agente é um velhote aparentemente doente.
Qual é a senha? GBD

'Bemvindo a Tóquio sr. Bond. Londres disse-me que estava para chegar.'
Deseja uma lista dos agentes que estão aqui? Sim
6 7 22 43 67 94
Nada ouvi dizer sobre esses planos — nem um rumor!

Você está ainda em Tóquio.
O seu contacto? 7

O agente é uma jovem elegante e sensual.
'Nada sei, sr. Bond — é verdade!'
Acredita? Não.
7 diz suavemente, 'Os únicos planos que tenho, James, pode consultá-los à sua vontade...'
Ela beija-o arrebatadamente
Você corresponde? Sim
Oh! Ela era uma agente russa. Enquanto você dormia, ela cortou-lhe o pescoço (gentilmente, claro...)

Perdeu! Era 23 quem tinha os planos.
READY

ÍNDICE